Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América
Facultad de Ingeniería Geológica, Minera, Metalúrgica y Geográfica
Escuela Profesional de Ingeniería de Minas

Optimización de la ley en el secuenciamiento del remanejo de stockpiles usando programación lineal entera mixta y Visual Basic

TESIS

Para optar el Título Profesional de Ingeniero de Minas

AUTOR

Manuel Alejandro CLEQUE VALVERDE

ASESOR

David Ysaac MELGAR CABANA

Lima, Perú

2020

Hoja de metadatos complementarios

Código ORCID del autor	---
DNI o pasaporte del autor	48351287
Código ORCID del asesor	https://orcid.org/0000-0001-6996-923X
DNI o pasaporte del asesor	41423051
Grupo de investigación	---
Agencia financiadora	---
Ubicación geográfica donde se desarrolló la investigación	Lima – Lima – Perú S 12°2'35.4" O 77°1'41.7"
Año o rango de años en que se realizó la investigación	2018 – 2020
Disciplinas OCDE	Geociencias, Multidisciplinar http://purl.org/pe-repo/ocde/ford#1.05.01

Referencia bibliográfica

Cleque, M. (2020). *Optimización de la ley en el secuenciamiento del remanejo de stockpiles usando programación lineal entera mixta y Visual Basic*. Tesis para optar el título de Ingeniero de Minas. Escuela Profesional de Ingeniería de Minas, Facultad de Ingeniería Geológica, Minera, Metalúrgica y Geográfica, Universidad Nacional Mayor de San Marcos, Lima, Perú.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
(Universidad del Perú - Decana de América)
FACULTAD DE INGENIERÍA GEOLÓGICA, MINERA, METALÚRGICA Y GEOGRÁFICA
ESCUELA PROFESIONAL DE INGENIERÍA DE MINAS
Av. Colonial cdra. 53 – Ciudad Universitaria
Central Telefónica: 619-7000 anexos: 1110 - 1111
Lima 1 – Perú

ACTA DE SUSTENTACIÓN DE TESIS PARA OPTAR TÍTULO PROFESIONAL DE INGENIERO DE MINAS

En las instalaciones de la Escuela Profesional de Ingeniería de Minas de la Facultad de Ingeniería Geológica, Minera, Metalúrgica y Geográfica de la Universidad Nacional Mayor de San Marcos, el jueves 05 de noviembre del 2020, siendo las 15:00 horas, en presencia de los Señores Docentes designados como Miembros del Jurado.

Mg. ALFONSO ALBERTO ROMERO BAYLÓN	*Presidente*
Dr. JORGE ENRIQUE SOTO YEN	*Miembro*
Dr. ARÍSTIDES SOTOMAYOR CABRERA	*Miembro*

Reunidos para el Acto Académico Público de la Sustentación de la TESIS de ***Don MANUEL ALEJANDRO CLEQUE VALVERDE****, Bachiller en Ingeniería de Minas, quien sustentó la Tesis Titulada:* ***"OPTIMIZACIÓN DE LA LEY EN EL SECUENCIAMIENTO DE REMANEJO DE STOCKPILES USANDO PROGRAMACIÓN LINEAL ENTERA MIXTA Y BISUAL BASIC"****, para la obtención del Título Profesional de Ingeniero de Minas.*

Los miembros del Jurado Calificador, escuchada la sustentación respectiva, plantearon al graduando las observaciones pertinentes, que fueron absueltas a:

APROBADO

.....

El Jurado procedió a la calificación, cuyo resultado fue la nota de:
15 (QUINCE)

.....

Habiendo sido aprobada la Sustentación de la Tesis por el Jurado Calificador, el Miembro Presidente del Jurado, recomienda que la Facultad de Ingeniería Geológica, Minera, Metalúrgica y Geográfica, otorgue el ***TÍTULO PROFESIONAL DE INGENIERO DE MINAS****, a* ***Don MANUEL ALEJANDRO CLEQUE VALVERDE****.*

Siendo las 16:03 horas, se dio por concluido el acto académico, expidiéndose cinco (05) Actas Originales de la Sustentación de Tesis.

Ciudad Universitaria, 05 de noviembre del 2020

Mg. ALFONSO ALBERTO ROMERO BAYLÓN
MIEMBRO PRESIDENTE

Dr. JORGE ENRIQUE SOTO YEN
MIEMBRO

Dr. ARÍSTIDES SOTOMAYOR CABRERA
MIEMBRO

Mg. ENRIQUE GUADALUPE GOMEZ
DIRECTOR
ESCUELA PROFESIONAL DE INGENIERÍA DE MINAS

RECOMENDACIONES

Datos de la plataforma virtual institucional del acto de sustentación:
ID: https://meet.google.com/ien-nvmo-iyx
Grabación archivada en: https://drive.google.com/file/d/1vbbFo1iU7v2652x27dbirMMDFyR_CHkp/view

NOTA OBTENIDA: 15 (Quince)

PÚBLICO ASISTENTE: (Nombre, apellido y DNI)

1. Jaquelyne Ames Sovero DNI 71561803
2. Jesus Camarena Ames DNI 08102745
3. Yajaira Nohelia Mosqueira Mostacero DNI 46421665
4. Manuel Reynaldo Montenegro Perez DNI 72213886
5. Edison Herbas Berrospi DNI 74253640
6. Laura Gabriela Cabrera Roque DNI 42785117
7. David Ysaac Melgar Cabana DNI 41423051
8. Roy Inga Aguilar DNI 74587122
9. Ricardo Vera Huamani DNI 48304937
10. Johnny Alberto Rondán Ricaldi DNI 47582945
11. Elmer Antonio Perez Zambrano DNI 46911300
12. Luis Vicente Morales Soto DNI 40080376
13. Alcides Peña Arostegui DNI 73201433
14. Brayan Sullca Pinares DNI 72689408
15. Alexander Rivera Retamozo DNI 72077553
16. Max Suarez Cornelio DNI 74491920
17. Eduardo Rojas Dyer DNI 08261630
18. Calixto Romero Valdez DNI 46056687
19. Jhor Keven Oscco Cardenaz DNI 71008948
20. Said Cardenas Huallpa DNI 73859154
21. Sarita Valverde Becerra DNI 47644470

Dedicatoria

Este trabajo se lo dedico a Dios, por estar a mi lado siempre. A mis padres, Blanca y Martin y a mi hermano Adrián por ser el motor de mi vida. Y también a mi asesor, el ingeniero David Melgar, quien me apoyo con la revisión y actualizaciones de la presente investigación. A Pabel Alvarado por brindarme el apoyo con la licencia del software.

Gracias a todos

Resumen

Un plan de extracción de stockpiles debe realizarse de manera que cumpla los objetivos establecidos y considere las restricciones a las cuales se enfrenta la operación minera. En este caso, el objetivo es maximizar la ley de plata que se enviará a la planta de procesos y la principal restricción será mantener la ley de cobre que se envía a planta dentro de un rango aceptable para que no perjudique la recuperación metalúrgica. El cobre que se obtenga no es comercializado por política de la empresa.

El ejercicio numérico realizado en este estudio consistió en definir: la ley de corte para clasificar el mineral y el desmonte, generación de polígonos para secuenciar teniendo como unidad de planificación polígonos de 2,500 toneladas, secuenciamiento de extracción de polígonos utilizando un modelo de programación lineal mixta entera en la cual la función objetivo debe ser maximizar la ley de plata y las restricciones son las leyes y tonelajes máximos y mínimos que puede aceptar la planta concentradora. Finalmente, se realizó una evaluación económica para calcular los ingresos obtenidos del plan de extracción propuesto.

Se obtuvo un plan de extracción para doce meses durante los cuales se extrae 964,520 toneladas de mineral con una ley promedio de plata de 6.31 oz/ton y 107,969 toneladas de desmonte. Los primeros ocho meses se logra cumplir con las 75,355 toneladas de mineral mensuales con una ley de plata promedio de 6.67 oz/ton y una ley de cobre menor a 0.125%. Los últimos cuatro meses del programa de extracción presenta un importante incremento de la ley de cobre debido a la culminación de los stocks de baja de ley de cobre.

Finalmente, en la evaluación económica, este plan nos genera unos ingresos totales de 19.6 millones de dólares por los doce meses.

Índice

Índice de Figuras

Índice de Tablas

Capítulo I

Planteamiento de estudio

1.1. Planteamiento del problema

Una unidad minera ubicada en el centro del país posee ocho pilas de mineral y quiere generar un plan de extracción mensual combinando dichas pilas de mineral con el objetivo de obtener la ley de plata máxima y mantener la ley de cobre dentro de un rango de ley aceptable considerando la capacidad de tonelaje mínima y máxima que pude procesar la planta mensualmente. Los softwares mineros actuales están enfocados a secuenciar operaciones mineras teniendo como punto de origen el tajo abierto y como puntos de destino la planta concentradora y/o pad de lixiviación, sus respectivos stocks y los botaderos de desmonte, entre otros destinos. Sin embargo, el problema que se plantea en la presente investigación es un caso atípico donde sucede lo contrario, los stocks en lugar de constituir destinos constituyen orígenes. Así mismo, por lo general, los modelos matemáticos de estos softwares tienden a estar ideados para tener más puntos de destino

que puntos de origen, y en este problema sucede lo contrario. Por tal motivo, encontramos que al usar dichos programas no obtenemos los resultados esperados rápidamente ni obtenemos la versatilidad que esperaríamos.

Para el presente caso se considerará un mínimo de 70,000 toneladas mensuales y un máximo de 80,0000 toneladas mensuales.

1.1.1. Problema principal

¿Se puede generar un plan de extracción mensual de ocho stockpiles maximizando la ley de plata?

1.1.2. Problemas específicos

1) ¿Son correctos los parámetros de entrada (modelo de bloques, superficie de stocks, topografía) que recibimos para realizar el secuenciamiento?
2) ¿Cómo diferenciamos el mineral del desmonte dentro de los stockpiles?
3) ¿Cómo agrupamos los bloques que pertenecen al mismo rango de ley de plata?
4) ¿Cómo obtenemos polígonos de secuenciamiento homogéneos en tonelaje y calidad de mineral?
5) ¿Cómo podemos aplicar el modelo matemático en el inventario de polígonos generado?
6) ¿Son favorables los resultados económicos obtenidos?
7) ¿Cómo podemos visualizar los resultados gráficamente?

1.2. Objetivos de la investigación

1.2.1. Objetivos generales

Optimizar la ley en el secuenciamiento de la recuperación de stockpiles usando programación lineal entera mixta y Visual Basic, para obtener un secuenciamiento más rápido, versátil y demostrado con fórmulas matemáticas.

1.2.2. Objetivos específicos

1) Verificar los parámetros de entrada que se reciben, los cuales son: modelo de bloques, superficie de stockpiles y topografía.
2) Calcular la ley de corte para clasificar los stocks.
3) Generar polígonos según el tipo de material.
4) Dimensionamiento de polígonos para el secuenciamiento y mezcla óptima.
5) Cargar el inventario de polígonos dentro del programa que contiene el modelo matemático.
6) Realizar un análisis económico de los resultados del plan obtenido.
7) Exportar los resultados en formato dxf para visualizarlos en otros programas.

1.3. Hipótesis

1.3.1. Hipótesis general

Es posible realizar el secuenciamiento de la recuperación de stockpiles usando programación lineal entera mixta y Visual Basic.

1.3.2. Hipótesis específicas

1) Los parámetros de entrada recibidos son correctos y serán útiles para realizar el trabajo.

2) La ley de corte la calcularemos usando los parámetros de entrada.
3) Se pueden generar polígonos según el tipo de material.
4) Dimensionaremos los polígonos de manera que pueda secuenciarse correctamente y obtener una mezcla óptima.
5) Cargaremos el inventario de polígonos al programa que contiene el modelo matemático.
6) Realizaremos un análisis económico del plan obtenido utilizando parámetros económicos y técnicos.
7) Cargaremos los cortes por mes a un software de diseño para ver el plan visualmente.

1.4. Metodología de la investigación

La presente investigación fue diseñada bajo una metodología de proyectos factibles la cual se va a asentar sobre la explicación de los procedimientos necesarios para las actividades pensadas, el análisis de los recursos disponibles y las restricciones a las cuales está sometido el sistema que se está analizando.

En concreto, el diseño metodológico consistirá en realizar ciertas modificaciones al algoritmo de programación lineal entera mixta para adaptarlo al problema que está siendo objeto de estudio en la presente investigación. El siguiente paso será la implementación de dicho algoritmo en un programa computacional utilizando Visual Basic.

Finalmente, se realizará un análisis económico de los resultados obtenidos para evidenciar la factibilidad del proyecto.

1.5. Identificación y operacionalización de variables

Las variables relevantes en la presente investigación son las siguientes:

- Densidad
- Ley de plata
- Ley de cobre
- Recuperación metalúrgica de la plata
- Precio de la plata

Las primeras tres variables las encontramos dentro del modelo de bloques empleado para el cálculo. La densidad es una variable importante debido a que de ella depende el tonelaje que contiene cada stock. En el caso de tener bloques sin densidad asignada, se preparó el programa para asumir un valor por defecto de 2.5 ton/mc. La ley de plata nos servirá para poder clasificar el material dentro de cada stock, este valor está expresado en onzas por tonelada. La ley de cobre es un valor restrictivo para el modelo matemático planteado, es decir, se buscará mantener su valor total en los periodos dentro de unos límites determinados previamente. Esto debido a que se trata de un contaminante que genera una disminución en la recuperación metalúrgica en la planta concentradora. La recuperación metalúrgica de la plata en la presente investigación tendrá un valor fijo expresado en porcentaje, lo mismo que el precio de la plata expresada en dólares por onza.

1.6. Limitaciones de la investigación

El modelo matemático planteado para resolver el problema presentado es un modelo de programación lineal mixta entera. Este modelo toma como función objetivo,

la suma de contenido metálico expresado en onzas de plata de cada polígono de 2,500 toneladas de cada stock. Debido a esto, tratar de plantear un modelo para cada periodo de tiempo (en este caso se tratará de periodos mensuales) es relativamente fácil, sin embargo, al momento de implementar esta cláusula en un programa como es el Visual Basic de Microsoft Excel es cuando la tarea se vuelve muy tediosa, incrementando el cálculo en n periodos.

No se presenta comparaciones del método empleado con los resultados de los softwares de otras empresas al no contar con el consentimiento de las mismas.

1.6.1. Problema del bloque

Al momento de crear el algoritmo de secuenciamiento debemos tener claro si se busca secuenciar bloque por bloque del modelo o si se realiza secuenciar por polígonos. La ventaja de realizar el secuenciamiento bloque por bloque es tener una mayor libertad para el secuenciamiento, sin embargo, al implementar el modelo matemático bloque a bloque aumentaría en 132 veces a hacerlo por polígonos, lo que se traduce en un mayor tiempo de procesamiento. Por esta razón es que para la presente investigación se ha decidido trabajar con polígonos de 2,500 toneladas para realizar el secuenciamiento.

1.6.2. Algoritmo multiperiodo

El modelo matemático planteado para resolver el problema presentado es un modelo de programación lineal mixta entera. Este modelo toma como función objetivo, la suma de contenido metálico expresado en onzas de plata de cada polígono de 2,500 toneladas de cada stock. Debido a esto, tratar de plantear un modelo para cada periodo de tiempo (en este caso se tratará de periodos mensuales) es relativamente fácil, sin embargo, al momento de implementar esta premisa en un programa como es el Visual Basic de

Microsoft Excel es cuando la tarea se vuelve muy tediosa, incrementando el cálculo en n periodos. Además, al incluir variables multi-periodo dentro del algoritmo restringe al usuario la capacidad de poder realizar cambios en los criterios fijados para el secuenciamiento, lo que se traduce en una disminución de versatilidad para poder secuenciar periodo a periodo. Por estas razones para la presente investigación, se planteará el secuenciamiento periodo por periodo.

Capítulo II

Generalidades

2.1. Softwares especializados en minería

"Diversas alternativas ofrecen la industria de los softwares mineros en el mercado nacional e internacional, desde soluciones orientadas a aplicaciones específicas hasta productos que ofrecen servicios más integrales, que abarcan y controlan diversas áreas del negocio minero". (Tutor Minero, 2010)

Los últimos años, las empresas desarrolladoras de softwares mineros han multiplicado sus esfuerzos en generar productos que faciliten al usuario secuenciar sus diseños para generar planes más realistas. Para ello, se han apoyado en muchas teorías y algoritmos tales como la programación lineal, programación no lineal, algoritmo de Minimax, e incluso simulación de Monte Carlo, entre otros.

La minería, como todos los sectores productivos de la sociedad, recurre a las herramientas tecnológicas difundidas y utilizadas actualmente, tal como el software. Actualmente se utiliza el software, desde el más básico hasta el más complejo, en todos

los procesos, empezando por el análisis de datos, presentación de los mismos hasta llegar a los modelos predictivos estocásticos como la simulación de procesos minero-metalúrgicos y como herramienta de gestión en las empresas mineras. (Guzman & Ortiz, 2014)

Se pueden observar dos corrientes en relación al desarrollo de software actualmente, por lo menos a nivel de Latinoamérica.

Una de ellas es la tendencia a complejizar los programas informáticos enfocados a minería cada vez más, a tal punto que se vuelven unos "superprogramas", los cuales requieren de "supercomputadoras" con una alta velocidad de procesamiento de datos, mejor capacidad de soportar gráficos pesados y discos duros de más de 2 TB de memoria. Todo esto con el fin de obtener la mayor cantidad de datos posibles, simulando vastos y complejos escenarios; sin embargo, muchas veces los profesionales que obtenemos estas ingentes cantidades de información no sabemos cómo analizarla, filtrarla, interpretarla y darle un fin útil, y esto está relacionado con el fin que busca la otra corriente de desarrollo de software.

La otra corriente se centra en desarrollar programas mucho más sencillos, de entendimiento rápido por parte del usuario. Estos programas, generalmente, están concebidos para dar asistencia a operaciones no muy complejas, las cuales manejan no más de dos variables. El entendimiento de la base teórica que se aplica en un software es el pilar mayor de los profesionales y técnicos que apoyan esta corriente.

2.2. Administración de la información

El manejo de la información es una de las actividades importantes en la investigación. Nosotros manejamos dos documentos importantes; el modelo de bloques y la topografía actual donde está ubicado el proyecto objeto de estudio.

El modelo de bloques lo manejamos en un formato de valores separados por comas (o también llamados archivos CSV), el cual es un tipo de documento en formato abierto que almacena la información en tablas, en las que las columnas se separan por comas. Al tener el modelo de bloques en este formato nos permite una rápida depuración y valoración de la misma a través del programa Microsoft Excel.

Otro de los beneficios de manejar el modelo de bloques en este tipo de archivos es la eficiencia en espacio de almacenamiento, este tipo de archivos tiene un peso menor al clásico MS-DOS, sólo siendo superado en este aspecto, en algunas ocasiones, por los archivos de texto (.txt).

Para el manejo de la topografía actual, usamos el formato dxf (acrónimo del inglés Drawing Exchange Format), el cual es un formato de archivo para dibujos de diseño asistido por computadora.

Este tipo de formato es muy usado para manejar archivos de diseño y topografías debido a su compatibilidad con todas las versiones del AutoCAD.

Existen otras maneras de manejar la información como podría ser a través de una base de datos relacional, o no relacional, o utilizando la nube para alojar la información. Para implementar una base de datos, requerimos de un gestor de base de datos.

Respecto a los gestores de base de datos, existen los que son de licencia libre y los de licencia de pago. Entre los gestores con licencia libre se encuentran: MySQL, SQLite3, PosgreSQL, entre otros.

El gestor de bases de datos más utilizado es el Microsoft SQL Server, el cual no es de licencia libre.

A pesar que son variados los gestores de bases de datos, el lenguaje para crear, consultar, modificar, actualizar y borrar una base de datos es similar en todos los gestores de base de datos, siempre y cuando estos manejen bases de datos estructuradas.

Las bases de datos estructuradas y las bases de datos no estructuradas no pueden ser manejadas ni almacenadas en un mismo gestor de base de datos. Estos difieren mucho en su estructura y el lenguaje para realizar consultas.

Finalmente, para la presente investigación no se vio la necesidad de implementar una base de datos debido a que se requeriría una sola tabla sin relaciones ni paso de parámetros de una tabla a otra.

2.3. Algoritmos de secuenciamiento

El planeamiento minero es un proceso muy importante dentro de todo proyecto minero debido a que definiendo el valor de la reserva minera establece el valor económico de todo el proyecto en general. Siendo más específicos, muchas veces no se contemplan todos los riesgos asociados al proyecto tales como el nivel de detalle de la información, la fluctuación de los precios entre otros. Encontrar el plan minero óptimo para desarrollar el minado es un problema muy complejo debido a la existencia de múltiples variables en el cálculo, teniendo que analizar múltiples escenarios. (Amaya & Nancel-Penard, 2010)

El problema de la planificación minera puede ser formulado partiendo del hecho que se tiene una estimación de las distribuciones del valor mineral (leyes) in-situ, con lo cual el planificador debe asignar la secuencia espacio/temporal de extracción de bloques.

El objetivo de dicho secuenciamiento debe ser el maximizar el valor del negocio a largo plazo. Además, debe cumplir con las restricciones impuestas al sistema, tales como: capacidad de extracción, capacidad de planta, ángulos de talud, relación estéril/mineral, destinos.

Existen muchos algoritmos que plantean soluciones para el problema del secuenciamiento de minado; algunos son un poco antiguos como es el caso del algoritmo multiperiodo de Thys B. Johnson quien concibe el problema de secuenciamiento de la producción de una mina como un problema de programación lineal a gran escala usando como unidad elemental el bloque del modelo.

Johnson sostiene que la clave del éxito de su algoritmo radica en la identificación y cuantificación de las variables y restricciones principales que gobiernan el sistema minero a planificar. También destaca que se debe tener en cuenta en la fase en la cual se encuentra situado el proyecto minero o la mina; esto debido a que en etapas tempranas del proyecto minero la información con la que se contaría tendría una incertidumbre alta mientras que proyectos ya en desarrollo, que cuentan con registros históricos provenientes de distintas áreas de la mina, contarán con una incertidumbre menor.

Otro de los conceptos que Johnson busca superar es la sobrevaloración de la relación de desbroce, empleado en el método de secciones verticales, la cual según él presenta muchos sesgos al tratarse de un cálculo completamente económico o empírico.

Dos de los conceptos que empleó Johnson en su algoritmo y que perduran hasta el día de hoy son el concepto del bloque y la ley de corte para clasificar los mismos. Define el bloque como la unidad en la que se puede representar la mayoría de las actividades desarrolladas en una operación minera a tajo abierto, tales como: mallas de perforación cuadradas, carguío de material por parte de palas o cargadores de un determinado banco, transporte del mismo. Además, Jonhson en su algoritmo no solo se limita al secuenciamiento del tajo abierto sino, también aborda el secuenciamiento en la planta concentradora y en la refinería para lo cual asume que el bloque de minado va reduciendo su tamaño en cada etapa.

Un estudio realizado por Michellle Blom y Adrian R. Pearce titulado, "Short-Term Planning fo Open Pit Mines: A Review (Blom & R. Pearce, 2018), nos describe los algoritmos de planeamiento más importantes que se han empleado en la industria minera. Nos describe los primeros modelos de programación lineal para secuenciar, como es el caso de Topuz y Duan quienes empezaron a utilizar la investigación de operaciones en el planeamiento de minado. Particularmente, para el planeamiento a corto plazo, en 1985, se comenzaron a usar algoritmos basados en programación lineal para resolver el problema de mezclas en cada periodo de tiempo.

Michelle Blom, en su investigación, también nos presenta a Gershon M., quien propone dos algoritmos heurísticos para el secuenciamiento de minado aplicable en un contexto de corto plazo. El primero resuelve los problemas de mezcla tomando en cuenta

un periodo a la vez, seleccionando regiones a ser minadas en cada periodo, optimizando los objetivos definidos en las políticas de largo plazo. El segundo calcula cada bloque que puede ser minado después del que se viene minando ahora mismo, iterativamente seleccionará el bloque más adecuado para ser minado. La decisión la toma basándose en la posición del bloque y su calidad.

Blom también nos introduce PITSCHED, una herramienta interactiva diseñada por Fytas para ser aplicada al planeamiento a largo plazo y el planeamiento a corto plazo. El usuario agrupa los bloques del modelo en regiones de pala o carguío, asignándole una prioridad a cada región. Una programación lineal resuelve y calcula la fracción de cada región a ser minada en cada periodo, maximizando un objetivo diseñado para atender las zonas de mayor prioridad antes que las zonas de menor prioridad.

Sunder y Acharya en su investigación, Blast schedule planning and shiftwise production scheduling of anopencast iron ore mine, implementaron una modelo en programación lineal agrupando y seleccionando zonas de voladura en lugar de áreas de carguío.

Estudios más recientes como el de H. Askari-Nasab de la Universidad de Alberta propone un modelo de programación lineal entera mixta el cual busca secuenciar la extracción de los bloques en multiples periodos.

El algoritmo planteado por Askari toma una decisión en el periodo, destino y ruta que el mineral y el desmonte deben tomar con el fin de minimizar los costos y respetar las restricciones impuestas al sistema de minado y de procesos. El objetivo del algoritmo de Askari-Nasab es determinar la óptima secuencia de extracción de los bloques para cumplir los objetivos fijados por el largo plazo en cada periodo del corto plazo.

2.4. Aplicación de programación en minería

La programación cada vez viene aplicándose de manera más intensiva en todos los campos del conocimiento y todos los sectores socio-económicos la emplean para optimizar recursos y tiempo. Desde el año 2000 aproximadamente, el paradigma de programación cambió. Pasamos de una metodología de programación orientada a procedimiento a una orientada a objetos, esto cambió todo lo que se conocía hasta ese entonces acerca de los alcances y aplicaciones de la programación en la vida de los seres humanos.

Por lo tanto, con la metodología de programación orientada a objetos lo que se busca es representar todo aspecto de la vida real en código, digitalizar objetos es uno de los pilares de la programación actual predominante.

Para hablar sobre programación, tenemos que hablar sobre lenguajes de programación. El lenguaje de programación es un conjunto de sentencias escritas mediante código o símbolos que se usan para definir esquemas en el desarrollo de un software. (Rockcontent, 2019)

A través de este código se comunican el creador del programa y la computadora, especificando de forma clara los siguientes puntos:

- Los datos que utilizará el software
- La administración de los datos
- Los procedimientos o métodos que debe ejecutar el programa ante determinadas situaciones específicas

2.5. Las matemáticas en los softwares comerciales

Algunos softwares mineros son tan complejos y restringen al usuario al punto de volverse una especie de caja negra de la cual no se sabe cuales son los cálculos que realiza para conseguir determinados resultados.

Esto vuelve al usuario, por lo general, especialmente al ingeniero de minas, en un digitador de sentencia, comandos y rutinas, sin poder analizar la respuesta que esta obteniendo.

Esto restringe los potenciales que se pueden alcanzar si es que se conoce la base de los programas. Con esto nos referimos a los algoritmos matemáticos que son la base de todo programa, por ejemplo, el algoritmo que presentaré en la presente investigación, puede estar incluido en algún o en varios softwares comerciales, sin embargo, el poder entender el algoritmo me dio la posibilidad de implementarlo en el programa que mejor me parezca o al cual si tengo acceso.

Finalmente, en el desarrollo de programas es tan importante el diseño del código como el diseño del algoritmo o modelo matemático a implementar.

Capítulo III

Marco teórico

3.1. Antecedentes del estudio

Como se discutió en el capítulo anterior, una de los estudios más recientes acerca del tema objeto de investigación es el modelo de programación lineal entera mixta que propone Askari-Nasab.

Askari-Nasab resume el problema del secuenciamiento de bloques en dos interrogantes: cual bloque debe extraerse y cuando debe extraerse (siempre y cuando dicho bloque se encuentre dentro del límite del pit final). También debemos tener en cuenta las restricciones que recaen sobre el sistema minero, que además puede variar a lo largo del horizonte de tiempo a planear.

Askari originalmente en su estudio abarca el problema de secuenciar una mina a tajo abierto que posee diferentes destinos. En la Figura 1 se puede apreciar el esquema general del problema planteado por Askari-Nasab. Según la figura, existe un tajo abierto,

S stocks, P plantas de procesamiento y W botaderos de desmonte. La mina presenta E elementos, de los cuales, uno es considerado el elemento más importante.

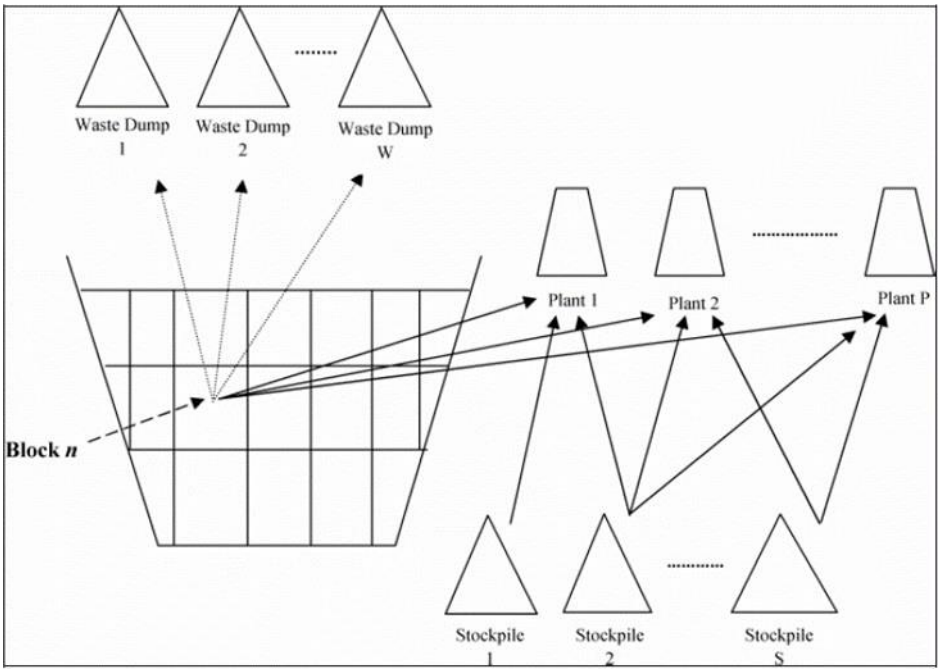


Figura 1. Esquema del problema

Askari sentencia que el motivo principal de manejar stocks dentro de la operación minera es para controlar el excedente del material minado del tajo para mantener el equilibrado la alimentación al área de procesos. Y cuando la producción de mineral del tajo abierto está por debajo de la capacidad de procesamiento, el mineral es recuperado de los stocks y enviado a la planta de procesos para cumplir con los objetivos de producción.

En su estudio, Askari-Nasab aplicó su algoritmo en un depósito de hierro. Para implementar dicho procedimiento utilizó TOMLAB/CPLEX como optimizador matemático.

El estudio de Askari-Nasab, al igual que la presente investigación, implementa el algoritmo tomando como unidad polígonos de minado en lugar de bloques del modelo en estudio.

3.2. Definición de términos

- **Concepto del bloque:** debido al tipo de equipos disponibles y sus capacidades, el minado de tajo abierto generalmente se realiza por bancos. La secuencia de minado generalmente consiste en disparar un volumen de material en un banco, cargar este material en un equipo de acarreo y transportar dicho material hacia un botadero de desmonte o a una planta concentradora. Las mallas de perforación usualmente tienen una forma rectangular y el material de un banco completo es cargado en una unidad. Por estas razones, una descripción bastante precisa de la operación minera puede darse en términos de unidades minables o bloques.

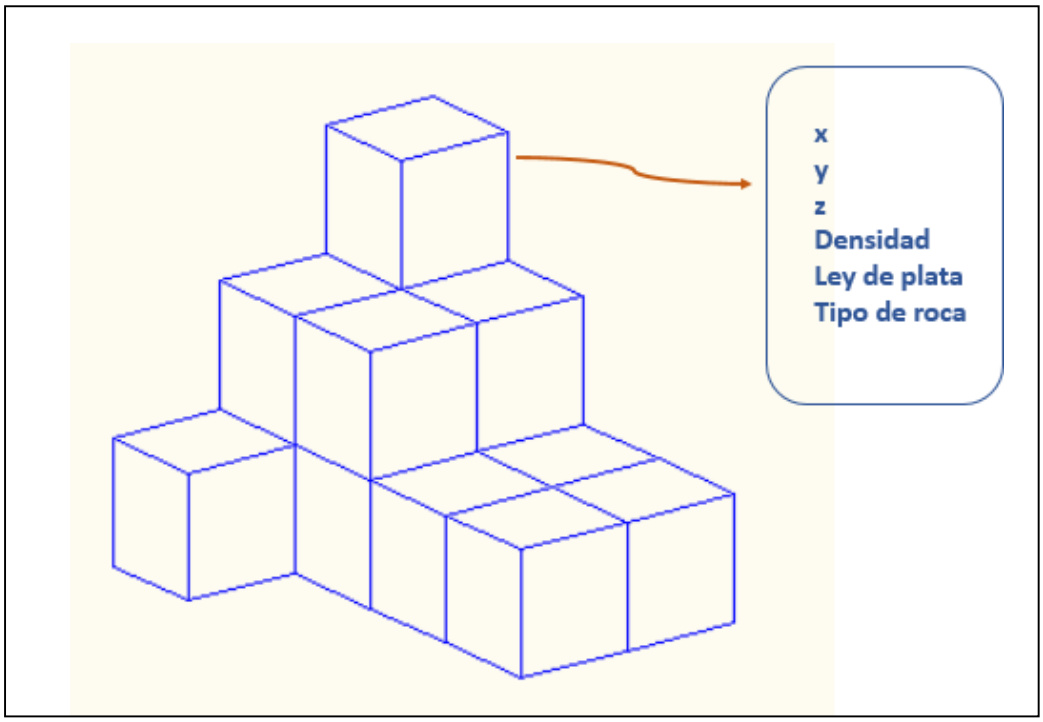


Figura 2. Vista esquemática del concepto del bloque

- **Ley de corte:** es un concepto económico el cual se usa generalmente para discriminar entre el mineral y el desmonte en un yacimiento mineral. Para el presente estudio se consideró una ley de corte estática en lugar de una ley de corte dinámica. La diferencia fundamental es que la ley de corte dinámica se expresa como una función del estado del sistema minero en el momento en que se debe tomar una decisión, así como los efectos futuros de la misma. El beneficio de una operación minera puede ser considerablemente determinada por la selección de la ley de corte y por lo tanto, esta es una decisión importante. La ley de corte económica que puede atribuirse al concepto del bloque ha reemplazado.
- **Pila de mineral:** son estructuras en forma de pirámide truncada o también pueden presentarse en forma de cono truncado. El material se almacena y se sustenta sobre sí mismo, formando un ángulo de reposo. Por conocimiento general, el ángulo de reposo de cualquier material es 37.5°, sin embargo, este ángulo puede ser elevado hasta 40° en algunos casos, especialmente cuando el material es de buena calidad y su humedad es mínima.

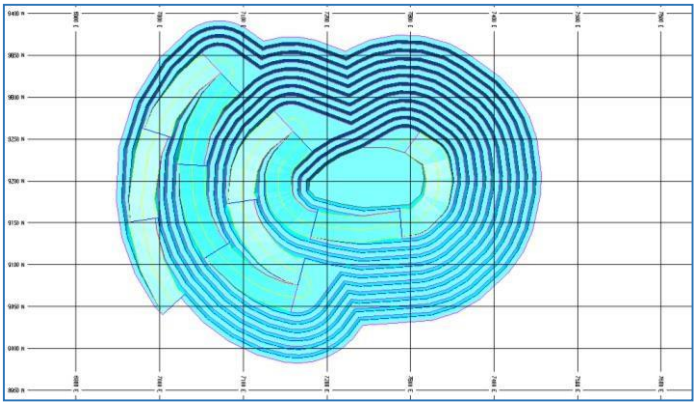


Figura 3. Diseño de pilas de mineral – vista en planta

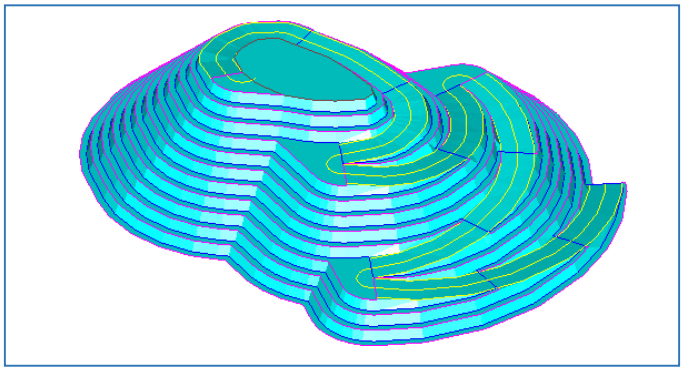


Figura 4. Diseño de pilas de mineral – vista isométrica

- **Mezcla óptima:** se considera mezcla óptima a la mezcla de material de distintos stocks que cumple con las restricciones impuestas por el sistema en el que se circunscribe además de conseguir el máximo beneficio del mismo. En este estudio, las restricciones se refieren a concentraciones máximas de contaminantes y las capacidades de planta y mina. Y el beneficio se consigue maximizando la ley del elemento valioso.

- **Función objetivo:** la función objetivo es el componente más importante en cualquier modelo de programación lineal puro o mixto. En el algoritmo de programación lineal, esta función es la que debe maximizarse o minimizarse, si embargo, también se deben respetar todas las restricciones impuestas en el sistema.

- **Variables de decisión:** representan los factores o variables controlables del sistema, sobre los cuales podemos tomar decisiones y asignarles un valor. En el modelo se busca saber el valor de estas variables para lograr conseguir el resultado óptimo fijado en la función objetivo. (Salazar Lopez, s.f.).

- **Restricciones:** son las limitaciones a las que están sujetas algunas de las variables que intervienen en el sistema. Representados generalmente por valores máximos y mínimos entro los cuales debe fluctuar los valores óptimos a encontrarse en la solución del algoritmo. (Salazar Lopez, s.f.).

3.3. Valorización del modelo de bloques

Un modelo de bloques bien elaborado debería contener toda la información relevante del material a ser minado tales como leyes, densidad, tipo de roca, alteración, y otros atributos que se consideren importantes para las diversas áreas interesadas en esta información.

La valorización de un modelo de bloques es necesaria para poder calcular el beneficio que se obtendrá del minado del mismo. En este estudio, la variable más importante para valorizar el modelo de bloques es la ley de plata. Esta variable se expresa en onzas por tonelada en el modelo de bloques objeto del presente estudio.

La forma más sencilla de valorizar el modelo de bloques es encontrando, en primer lugar, la ley de corte económica. Esta ley de corte económica nos permitirá discriminar los bloques que presentan una ley por encima de esta ley de corte como bloques de mineral mientras que los bloques que no cumplen este requisito deberán ser considerados bloques de desmonte.

Por lo tanto, la ley de corte económica representará aquella ley mínima con la cual el material (o el bloque) puede ser enviado a la planta concentradora (o al pad de lixiviación) obteniendo un beneficio económico. De otra forma, sería mejor enviar dicho bloque al botadero de desmonte y solo pagar el costo de minado del mismo.

En el supuesto que un bloque cuya ley de cabeza sea igual a la ley de corte es enviado como mineral, el beneficio obtenido por el procesamiento de este bloque sería exactamente igual a la suma del costo de minado y costo de procesamiento. Por esta razón, la ley de corte representa la ley con la cual no se gana ni se pierde valor económico al procesar el material.

Para calcular la ley de corte necesitamos los siguientes parámetros económicos, mineros e incluso metalúrgicos:

- Costo de minado ($/ton)
- Costo de proceso ($/ton)
- Gastos administrativos ($/ton)
- Precio ($/oz)
- Recuperación metalúrgica

Para explicar el procedimiento del cálculo de la ley de corte es necesario calcular los ingresos y costos proyectados para cualquier bloque del modelo. Los ingresos y los costos los calculamos con las siguientes fórmulas.

$$I = P\ x\ L\ x\ RM \ldots (1)$$

$$CT = CM + CP + GA \ldots (2)$$

Donde:

I: Ingreso ($/ton)

P: Precio de la plata ($/oz)

L: Ley de plata (oz/ton)

RM: Recuperación metalúrgica

CT: Costo total ($/ton)

CM: Costo de minado ($/ton)

CP: Costo de proceso

GA: Gastos generales y administrativos

Previamente explicamos que la ley de corte es aquella ley en la que los ingresos y los costos se equilibran. Por lo tanto, igualamos las ecuaciones 1 y 2.

$$I = CT \ldots (3)$$

$$P\ x\ L\ x\ RM = CM + CP + GA \ldots (4)$$

Observando la ecuación 4, nos damos cuenta que despejando todas las variables hacia la derecha y dejando al lado izquierdo solo la ley, esta ley será la ley de corte.

$$Lc = \frac{CM + CP + GA}{P\ x\ RM} \ldots (5)$$

Realizaremos el cálculo usado los parámetros planteados en la presente investigación.

$$\boldsymbol{CM} = 2.00 \frac{\$}{ton}$$

$$\boldsymbol{CP} = 5.50 \frac{\$}{ton}$$

$$\boldsymbol{GA} = 2.00 \frac{\$}{ton}$$

$$\boldsymbol{P} = 14.50 \frac{\$}{oz}$$

$$\boldsymbol{RM} = 14\ \%$$

Introduciendo estos parámetros en la ecuación 5. Obtenemos la siguiente ley de corte:

$$Lc = \frac{2.00 + 5.50 + 2.00}{14.50 * 0.14} = 4.68 \frac{oz}{ton}$$

En el Capítulo IV, Análisis de Resultados, veremos que el cálculo de la ley de corte también puede estar multiplicada por un factor, el cual protege la economía del proyecto respecto a las fluctuaciones en el tiempo del precio de mercado del metal que estamos proyectando extraer.

3.4. Polígonos de minado

"Los polígonos de minado son áreas dentro de cada banco que se caracterizan por presentar características específicas como tonelaje, tipo de material y ley promedio." (Herrada, 2007).

Estos polígonos se realizan con el fin de diferenciar el material del banco y darle la operatividad necesaria para que pueda ser minado por los equipos con los que se cuenta. Además, que es más fácil poder cubicar tonelajes, leyes, características del modelo de bloques a través del uso de polígonos.

Generalmente, en operaciones superficiales se utilizan equipos de gran dimensión por lo que resulta casi imposible poder realizar un minado bloque por bloque. Es por este motivo que se utilizan polígonos de minado.

Esta sección se ha dividido en dos partes: una llamada SMU la cual se centra en explicar la metodología que se debe utilizar para crear los polígonos con el fin de que sean operativos, disminuyan la dilución y además generen el mayor beneficio del modelo de bloques.

3.4.1. Unidad selectiva de minado

La definición convencional de Unidad Selectiva de Minado (SMU por sus siglas en inglés) es el mínimo volumen de material sobre el cual se puede clasificar en mineral o desmonte. (Sinclair & G. Blackwell, 2002)

En realidad, este concepto es más complejo. Si tuviéramos bloques de mineral de SMU en medio del desmonte, sería imposible e inoperativo poder extraer únicamente esos bloques de mineral. Asimismo, es imposible evitar extraer bloques de desmonte de SMU que se encuentran en medio del mineral. El tamaño del bloque SMU depende de varios

factores, tales como: tamaño de los equipos de mina, el método de minado a emplear, la dirección de minado, y el ambiente deposicional del cuerpo mineralizado.

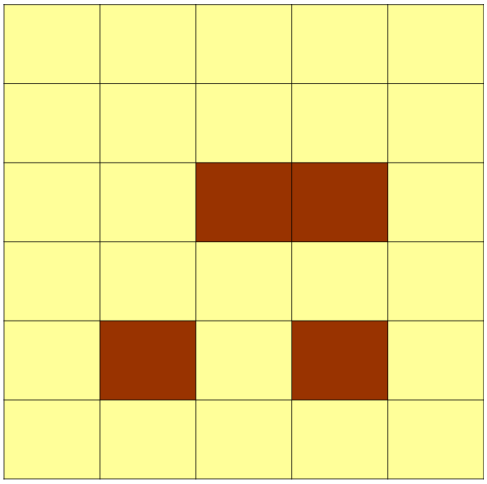


Figura 5. Bloques de desmonte en medio del mineral

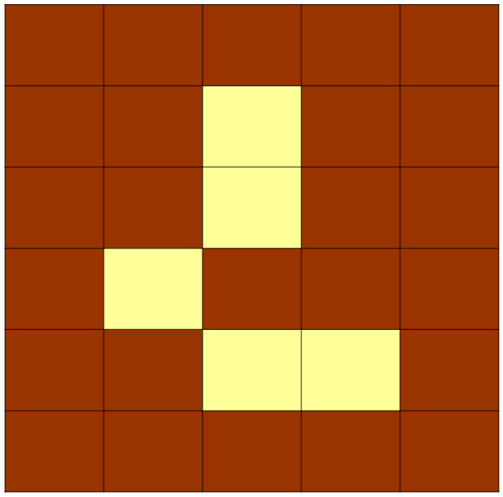


Figura 6. Bloques de mineral en medio del desmonte

La definición más correcta para SMU es el tamaño de modelo de bloques que podría predecir correctamente las toneladas de mineral, toneladas de desmonte, y la ley de cabeza diluida que la planta concentradora recibirá con la práctica anticipada del control de mineral. Este tamaño debe estar relacionado de alguna manera con la capacidad de los equipos para seleccionar material, pero también se basa en los datos disponibles para la clasificación (taladros de producción y/o taladros de control de mineral), los

procedimientos para traducir esos datos a límites de excavación explotables, y la eficiencia con que el equipo minero excava dichos límites. También se debe tener en cuenta diversas fuentes de dilución, incluida la dilución interna debido a la variabilidad de la ley dentro del mismo bloque SMU, la dilución externa resultante de los contactos geológicos que se modelan como superficies geométricas y la dilución operacional que tiene en cuenta los errores de producción, presiones y demandas del cronograma.

La práctica convencional de control de mineral usa la información proveniente de los muestreos de los taladros de producción y las inspecciones en campo para afinar los bordes de desmonte y mineral. Una forma común de trasladar los datos de los taladros de producción hacia los límites de excavación es usando el método del contorno y promedio donde las regiones de mineral y desmonte son delimitadas por un polígono que implícitamente representa a los equipos. El método del krigeaje es usado algunas veces para mejorar los bordes de desmonte y mineral. Los datos disponibles son una clara limitación a la resolución con los que podemos elegir los límites. Muestreos especiales de control de mineral o mallas de perforación más estrechas con taladros de menor diámetro proveen un dato más preciso, sin embargo, un análisis costo-beneficio se debe realizar.

En la práctica, las toneladas de desmonte, y las toneladas y la ley de mineral que recibe la planta concentradora es el resultado de un procedimiento de clasificación con muchos factores subjetivos. La planta concentradora ciertamente no recibe los valores en un modelo de largo o mediano plazo. Hay mucha más información al momento de minar y los bloques nunca se seleccionan libre y perfectamente en ninguna ocasión. No sería práctico con el software empleado y los recursos computacionales crear muchos escenarios y simular el procedimiento de clasificación en los múltiples modelos de alta

resolución. Debemos considerar la realidad del modelado de bloques para el tiempo presente.

3.4.2. Unidad de planificación

Unidad de planificación es un término que se buscó para poder lograr el secuenciamiento en la presente investigación. Para secuenciar la extracción de material, una vez determinado su destino, requerimos convertir el material de origen en variables de decisión que intervendrán en el algoritmo de programación lineal.

Por lo tanto, tenemos que segmentar el material que disponemos en polígonos de tal manera que cada polígono pueda representar una variable dentro del modelo matemático para la programación lineal mixta entera.

Existen muchas formas de determinar estos polígonos, sin embargo, para esta investigación se decidió utilizar el segmentador de polígonos que viene incorporado en el software Minero Suite. Minero Suite, como ya se mencionó anteriormente, es un software de diseño, optimización y planificación minera para operaciones a cielo abierto. Dentro de su módulo Smart Pit Designer, encontramos la herramienta para segmentar polígonos que es una herramienta muy fácil de usar ya que nos puede generar muchos polígonos en cuestión de segundos.

Esta herramienta del módulo Smart Pit Designer del programa Minero Suite nos solicita la siguiente información:

- Polígono a segmentar
- Punto inicial de segmentación
- Dirección de avance para segmentar
- Objeto de salida (donde se guardarán los polígonos)

- Área objetivo
- Radio

Los cuatro primeros valores de entrada los podemos seleccionar o señalar gráficamente en la misma interfaz del programa, mientras que los tres últimos deben ser digitados en una ventana que aparece durante la ejecución de la herramienta.

Por área objetivo, nos referimos a el área que debe tener cada polígono segmentado, es decir, si nuestro polígono de origen (polígono a segmentar) tuviera un área total de 400 metros cuadrados y nuestra área objetivo fuera de 50 metros cuadrados, la herramienta nos dará como resultado 8 polígonos de 50 metros cuadrados de área cada uno.

Esta herramienta supera a herramientas similares incorporadas en otros softwares de diseño de mina a tajo abierto debido a tratar de replicar el minado real de bancos. Por ejemplo, en otros programas la segmentación de estos polígonos lo realiza de una forma rectangular la cual no es muy precisa debido a que los equipos de minado usados en operaciones a tajo abierto no minan de forma rectangular ni cuadrangular sino de forma semicircular. Minero Suite con su segmentador si trata de reflejar lo mencionado líneas arriba porque sus segmentos los genera teniendo la forma semicircular como se mostrará en un ejemplo ilustrativo más adelante. Es por esta razón que el programa pide un valor para el radio, con este valor podrá calibrar el radio de circulo para generar cada polígono de una forma semicircular. Cabe resaltar que si introducimos valores incongruentes dentro de la ventana de la herramienta como serían un área demasiada extensa y un radio demasiado pequeño, no se podrá obtener resultados satisfactorios.

Con el fin de ser lo más didáctico posible en la presente investigación, mostraremos el procedimiento a seguir con el programa Minero Suite para poder segmentar un polígono dado.

En la Figura 7 presentamos la interfaz del programa Minero Suite mostrando el polígono a segmentar, en la barra lateral del lado izquierdo podemos observar el nombre del polígono que es poly.

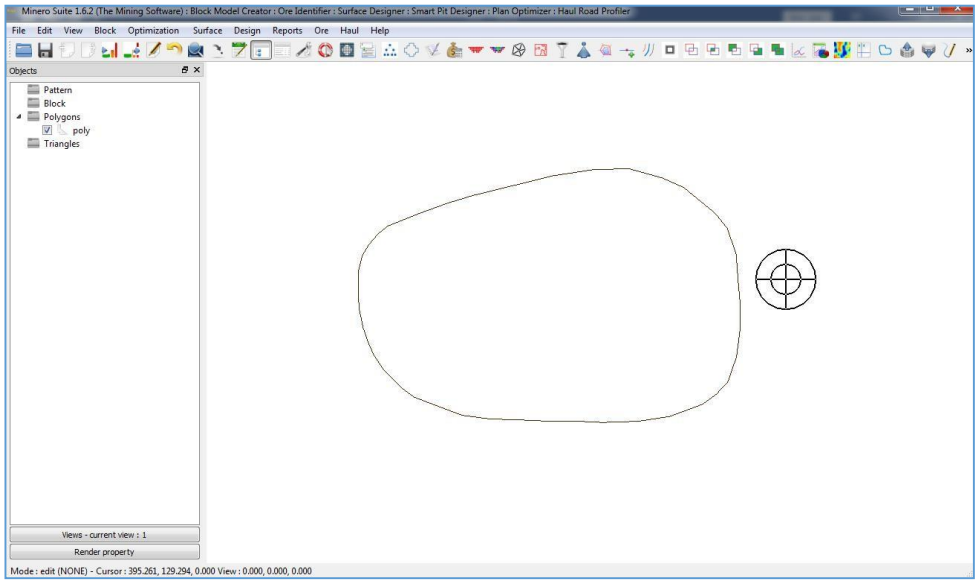


Figura 7. Interfaz de Minero Suite

Luego, tenemos que crear un objeto el cual será nuestro Objeto de salida. Para ello presionamos N y nos aparecerá la ventana que se muestra en la Figura 8 en la cual introduciremos el nombre del objeto el cual será poly2.

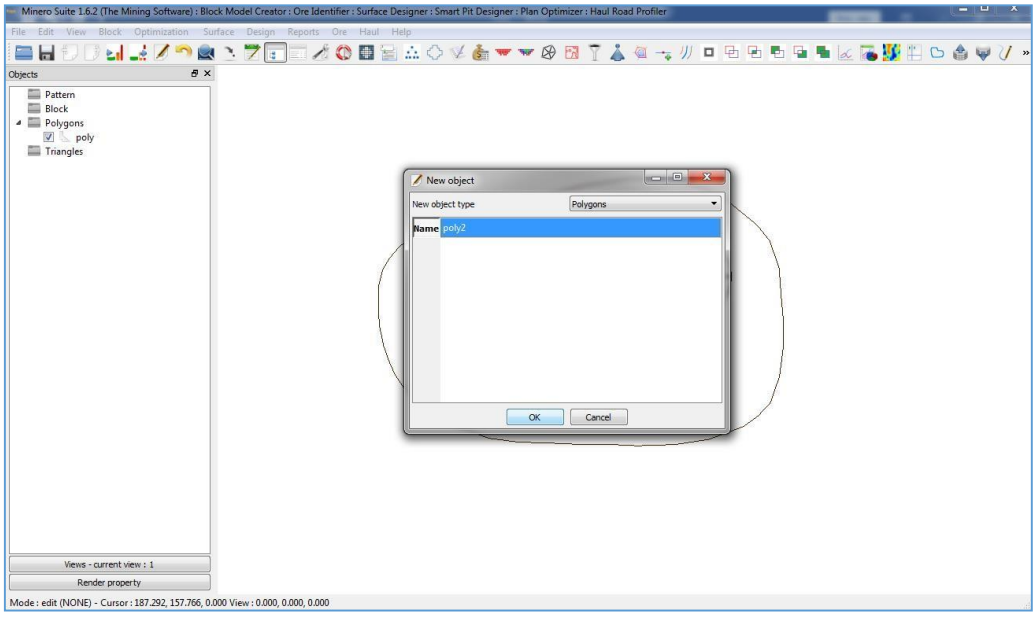


Figura 8. Ventana de diálogo para crear nuevo objeto en Minero Suite

Ahora, seleccionamos el polígono a segmentar, presionamos M y trazamos el punto de origen y la dirección de avance a segmentar como se muestra la línea azul en la Figura 9.

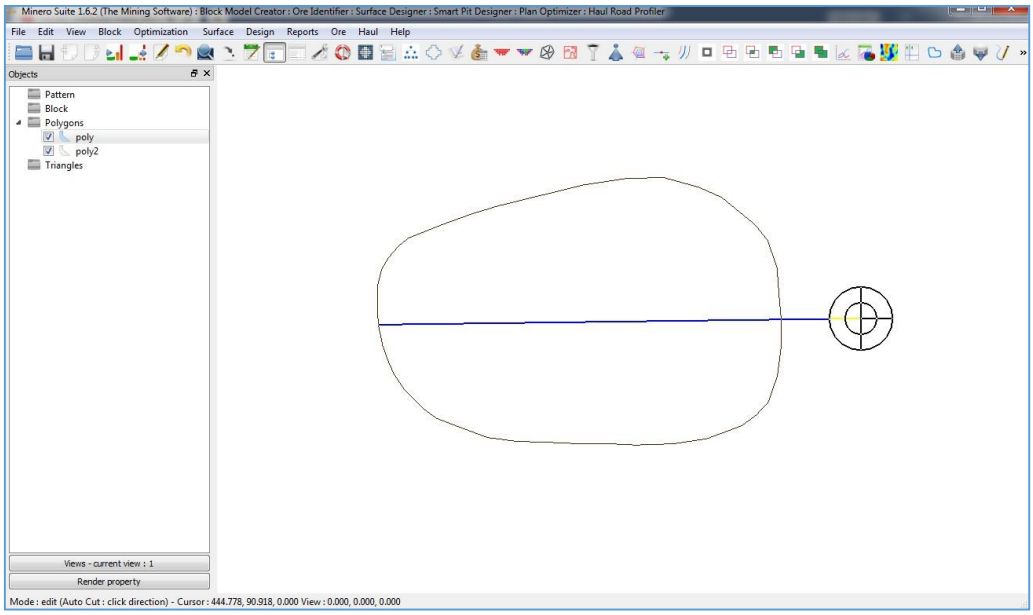


Figura 9. Polígono a segmentar con su dirección de avance

A continuación, presionamos la tecla R y nos aparecerá el cuadro de diálogo de la herramienta, como se muestra en la Figura 10, donde introduciremos los últimos valores de entrada que necesita la herramienta para ejecutar la tarea. El primer parámetro, que es el objeto de salida, ya fue generado y solo debemos seleccionarlo en la lista desplegable. El área total del polígono a segmentar es de 222,396 metros cuadrados, por lo que, si colocamos como área objetivo 30,000, el programa nos generará siete polígonos de 30,000 metros cuadrados y uno de 12,396 aproximadamente. Otro valor de entrada que necesita el programa es el radio, el valor introducido será de 450 metros para que nos dé como resultado los polígonos simétricos como se muestra en la Figura 11.

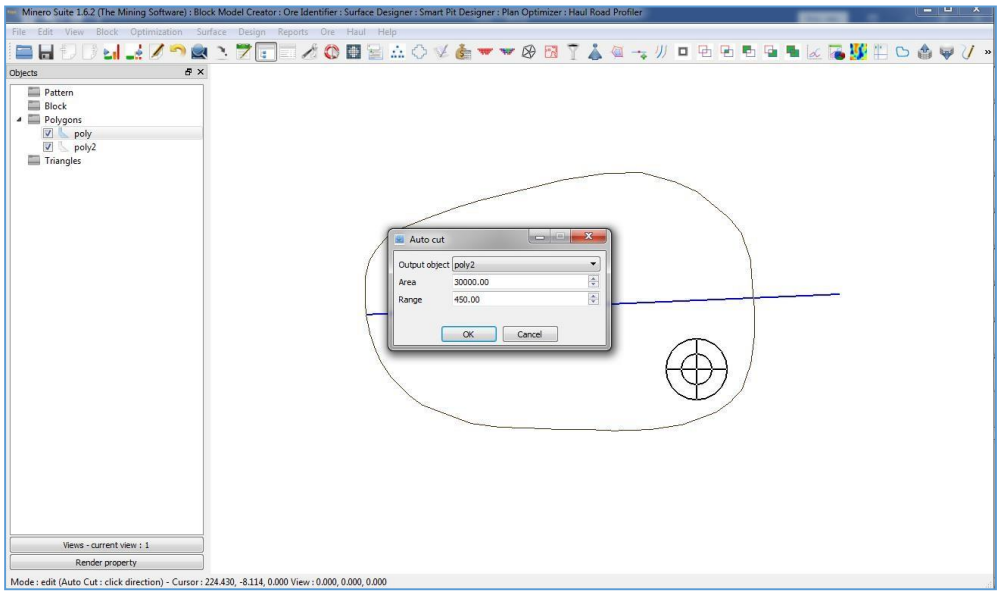


Figura 10. Ventana de diálogo de la herramienta para segmentar

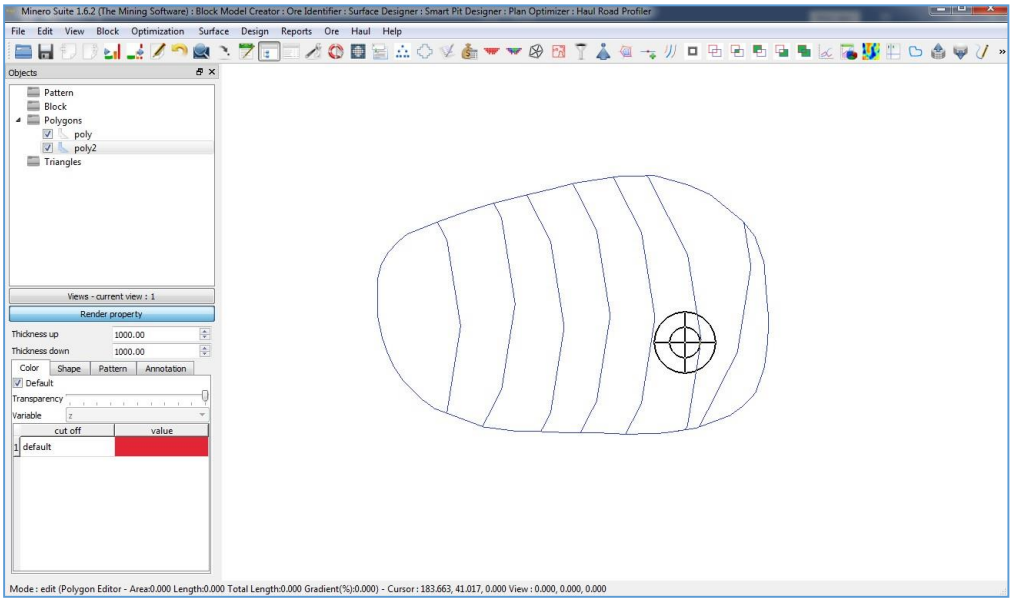


Figura 11. Objeto poly2 mostrando el polígono segmentado

Tener un área objetivo en lugar de un tonelaje o volumen objetivo es un problema porque generalmente nosotros queremos seccionar o segmentar por tonelajes o volúmenes, sin embargo, esta desventaja puede ser soslayada utilizando conceptos generales de geometría.

Por geometría general sabemos que el área puede estar relacionado con el tonelaje de un polígono. A continuación, presento un ejemplo ilustrativo para su mejor entendimiento.

Se busca obtener polígonos segmentados de 20,000 toneladas cada uno. Sabemos que la densidad promedio del polígono inicial es de 2.11 toneladas por metro cúbico (ton/m3). Por lo tanto, con la siguiente fórmula podemos convertir ese tonelaje objetivo en volumen objetivo.

$$Volumen\ objetivo = \frac{Tonelaje\ objetivo}{Desidad} = \frac{20{,}000\ ton}{2.11\ ton/m^3} = 9{,}479\ m^3$$

Ahora que tenemos un volumen objetivo, sabiendo que la altura del banco es de 10 metros, podemos convertirlo a un área objetivo con la siguiente fórmula.

$$Area\ objetivo = \frac{Volumen\ objetivo}{Altura\ de\ banco} = \frac{9{,}479\ m^3}{10\ m} = 947.9\ m^2$$

Como se presentó líneas arriba, el pequeño inconveniente de tener un tonelaje o volumen objetivo en lugar de un área puede ser superado aplicando estas simples fórmulas.

3.5. Programación lineal

La programación lineal es una técnica de modelización matemática desarrollada a partir de los inicios de la década de los 30's. Desde ese momento, se ha aplicado en muchas áreas de conocimiento de la humanidad, tales como: milicia, industrial, manufactura, medicina, genética, y también en mi campo de estudio que es la ingeniería de minas.

La programación lineal es un método de resolución de ecuaciones lineales utilizado para generar la mejor solución posible a un problema, el cual está sujeto a unos parámetros fijos que no pueden ser excedidos. (hiru.eus)

Un problema recurrente en minería de la programación lineal se muestra a continuación: teniendo n toneladas de un mineral con una ley de cabeza a y m toneladas de mineral con una ley de cabeza b. Cuantas toneladas de cada tipo de mineral deben enviarse para maximizar la ley, respetando las restricciones de tonelaje y ley impuestas al sistema.

3.5.1. Modelado

Las variables: son números reales mayores o iguales a cero.

$$X_i \geq 0$$

Las restricciones:

$$A_h = \sum_{l=1}^{N} a_{l,h}\, x\, X_l$$

$$B_h \leq \sum_{l=1}^{N} b_{l,h}\, x\, X_l$$

$$C_h \geq \sum_{l=1}^{N} c_{l,h}\, x\, X_l$$

Donde:

- **A:** valor establecido para no ser superado por exceso o defecto
- **B:** valor fijo que no puede ser superior
- **C:** valor que no puede ser diferente por exceso

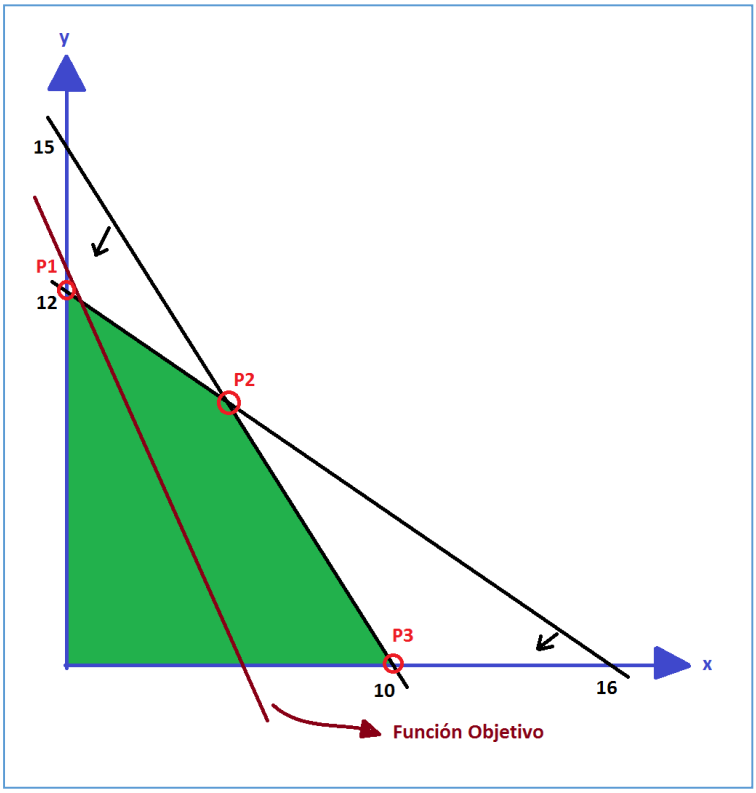


Figura 12. Programación lineal – método gráfico

- **h:** indice de la ecuación, variable de 1 a J (restricciones impuestas)
- **a,b,c:** coeficientes establecidos
- **x:** variables de 1 a N
- **l:** número de la variable, variable de 1 a N

Función objetivo

La función objetivo es la ecuación que será optimizada dadas las limitaciones o restricciones determinadas y con variables que necesitan ser minimizadas o maximizadas usando técnicas de programación lineal o no lineal. (Mora, 2016)

Solución factible

Las intersecciones que forman los semiplanos de las restricciones junto con la acotación fijada en las mismas o no, se denomina región válida o región donde se ubican las soluciones factibles.

Solución óptima

El vértice, del conjunto de vértices correspondiente a la región de soluciones factibles, que tiene la solución máxima o mínima al problema inicial se le denomina solución óptima. Pueden existir muchas soluciones factibles para el sistema, aquellas que cumplen con todas las restricciones impuestas; sin embargo, solo hay una que nos brinda el valor máximo, aquella que maximiza o minimiza la función objetivo.

3.5.2. Métodos de solución

Para resolver un algoritmo de programación lineal, existen muchos métodos de solución. La elección de alguno de ellos dependerá del número de variables que presente el sistema y el tamaño del sistema.

3.5.2.1. Método gráfico

Este método es una alternativa muy eficaz para poder obtener la solución de problemas de programación lineal cuando trabajamos con dos variables, donde la zona de soluciones factibles se encontrará en la zona positiva de los ejes coordenados X e Y. (Programacion lineal, 2018) .

Cuando graficamos las ecuaciones y la función objetivo y se define un área poligonal entre las rectas de las ecuaciones y los ejes cartesianos, se dice que existe solución factible y es precisamente la que se encuentra en los vértices de dicha región. Luego, el trabajo consiste en evaluar cada vértice como posible solución óptima y

determinar cuál de ellas satisface el requerimiento de la función objetivo, y esto dependerá si se trata de un problema de maximización o minimización. (Programacion lineal, 2018).

A continuación, se muestran los pasos requeridos para solucionar un problema de programación utilizando el método gráfico:

- Las inecuaciones son graficadas en el plano cartesiano, obteniendo el grupo de restricciones.
- Se grafica la función z como rectas paralelas que pasan por los vértices de las soluciones factibles anteriormente obtenidas.
- Se calcula el valor de la función objetivo en cada uno de los vértices anteriormente señalados. (hiru.eus)

Para explicar mejor este concepto, presento un sistema de programación lineal de dos variables y vamos a encontrar su solución con el método gráfico.

$Max\ z = 13x\ + 5y$

$15x + 10y \leq 150$

$12x + 16y \leq 192$

En la Figura 12 podemos visualizar la zona factible en color verde. En esta zona observamos todas las soluciones posibles al problema, sin embargo, en los vértices de la figura es donde encontraremos la solución que maximice la función objetivo.

En este problema, estos vértices están representados por los puntos P1, P2 y P3. Por lo tanto, reemplazaremos en la función objetivo cada punto y el que nos genere el mayor valor será la solución del problema.

$P1(0{,}12) \rightarrow 13\ x\ (0) + 5\ x\ (12) = 60$

$P1(4,9) \rightarrow 13\ x\ (4) + 5\ x\ (9) = 97$

$P1(10,0) \rightarrow 13\ x\ (10) + 5\ x\ (0) = 130$

El vértice P3 es el que maximiza la función objetivo del problema. El resultado del problema sería 130.

3.5.2.2. Método simplex

El método Simplex es un procedimiento iterativo que permite mejorar la solución de la función objetivo en cada paso. El proceso concluye cuando no se puede mejorar dicho valor, es decir, se ha alcanzado la solución óptima (la solución mayor en un problema de maximización o la solución menor en un problema de minimización). (phpsimplex)

Cuando le añadimos dos restricciones al sistema, el desarrollo del método simplex se facilita. El des (Taha, 2010).

- Las restricciones se configuran como ecuaciones con el lado derecho positivo.
- Las variables no pueden tomar valores negativos.
- Cuando tenemos desigualdades de este tipo, ≤, tenemos que hacer uso de la variable de holgura.

$3x_1 + 2x_2 \leq 20$

$3x_1 + 2x_2 + s_1 = 20$

La variable no negativa s1 es la holgura o cantidad no utilizada.

$x_1 + x_2 \geq 800$

$x_1 + x_2 - s_1 = 800$

El único requerimiento que nos falta es que el lado derecho de la ecuación resultante sea no negativo. Si el lado derecho resulta negativo, el requerimiento se satisface multiplicando ambos lados de la ecuación por -1.

Generalmente, el origen de coordenados, es decir, (0,0), es el método simplex lo que genera el valor de la función objetivo es igual a cero. La interrogante sería que un incremento de x1 y/o x2 puede incrementar el valor objetivo. Se tendrá que analizar la función objetivo:

$$Maximizar\ z = 2x_1 + 3x_2$$

Un aumento en alguna de las variables o en ambas en sus valores iniciales de cero incrementará el valor de la función objetivo. Presentamos un problema para graficar mejor la idea (Taha, 2010).

$$Maximizar\ z = 3x_1 + 4x_2$$

Sujeto a

$$6x_1 + 4x_2 \leq 24$$

$$x_1 + 2x_2 \leq 6$$

$$-x_1 + x2 \leq 1$$

$$x_2 \leq 2$$

$$x_1, x_2 \geq 0$$

Ahora, le damos forma al problema para solucionarlo aplicando el método simplex.

$$Maximizar\ z = 3x_1 + 4x_2 + \mathbf{0}\boldsymbol{s_1} + \mathbf{0}\boldsymbol{s_2} + \mathbf{0}\boldsymbol{s_3} + \mathbf{0}\boldsymbol{s_4}$$

Sujeto a

$$6x_1 + 4x_2 + \boldsymbol{s_1} = 24$$

$$x_1 + 2x_2 + \boldsymbol{s_2} = 6$$

$$-x_1 + x_2 + \boldsymbol{s_3} = 1$$

$$x_2 + \boldsymbol{s_4} = 2$$

$$x_1, x_2, \boldsymbol{s_1}, \boldsymbol{s_2}, \boldsymbol{s_3}, \boldsymbol{s_4} \geq 0$$

Las variables s1, s2, s3 y s4 son las holguras asociadas con las restricciones respectivas.

A continuación, escribimos la ecuación objetivo como:

$$z - 5x_1 - 4x_2 = 0$$

De esta manera, la tabla inicial simplex se representa como sigue:

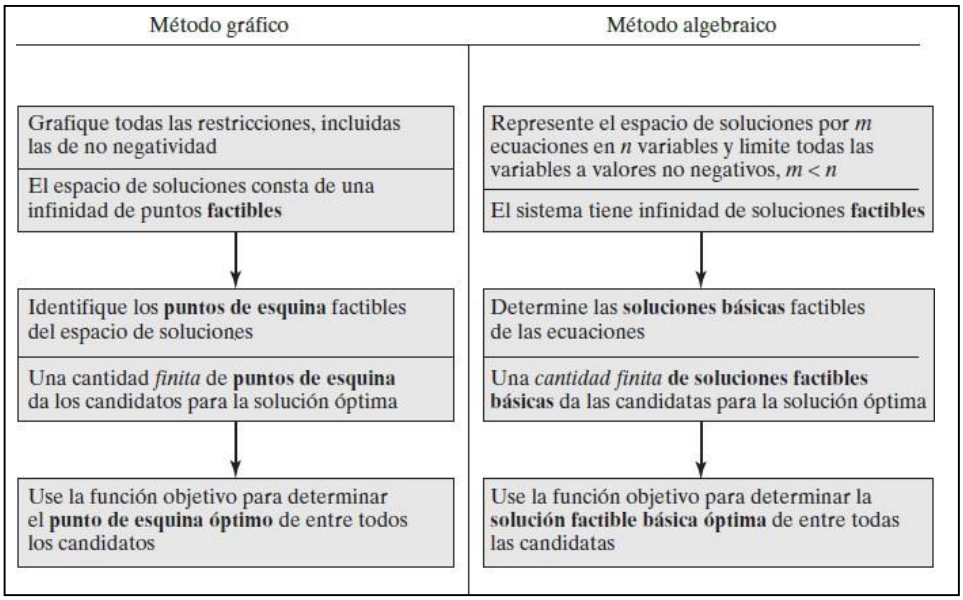


Figura 13. Comparación entre método gráfico y método algebraico

Tabla 1
Tabla inicial para el cálculo por el método simplex

Básica	Z	x1	x2	s1	s2	s3	s4	Solución	
Z	1	-3	-4	0	0	0	0	0	Fila z
s1	0	6	4	1	0	0	0	24	Fila s1
s2	0	1	2	0	1	0	0	6	Fila s2

s3	0	-1	1	0	0	1	0	1	Fila s3
s4	0	0	1	0	0	0	1	2	Fila s4

La primera tabla simplex nos da de manera automática la primera iteración. La función z y las variables principales se muestran en la parte extrema izquierda de la tabla. El resultado podría ser igual en las variables x1 y x2 en todo el sistema y concluyendo en este paso con la matriz identidad de los indicadores de las variables básicas.

La función z = 3x1 + 4x2 nos indica que la función puede incrementar su valor si se aumenta el valor de una de las variables no básicas. La incógnita x1, al tener un indicador más positivo, debe ser la variable que debe incrementar su valor porque genera más impacto en el valor de la función objetivo. De esta manera, la variable x1 se constituye como la variable de entrada del método simples.

La incógnita de entrada, una vez que se constituye como tal, se transforma en no básica. Para hallar la incógnita de salida se debe relacionar el extremo derecho de las restricciones restringiendo los valores positivos bajo la incógnita de entrada.

Básica	*x1 entrante*	*Solución*	*Relación (o intersección)*
s1	6	24	$x_1 = \frac{24}{6} = 4$ ← mínimo
s2	1	6	$x_1 = \frac{6}{1} = 6$
s3	-1	1	$x_1 = \frac{1}{-1} = -1$ ← (denominador negativo, ignorar)
s4	0	2	$x_1 = \frac{2}{0} = 00$ ← (denominador cero, ignorar)

Figura 14. Esquema mostrando el cálculo en la tabla inicial simplex

Conclusión: la variable x1 se configura como variable de entrada (en el cuarto nivel) y la incógnita x2 es la que sale (en el primer nivel)

Se observa que las intersecciones de las restricciones en el plano cartesiano se muestran en la figura 15. El valor de la incógnita x1 aumenta hasta el cruce en el lado no negativo del plano con el eje de la primera variable hasta encontrar el vértice B. No está permitido un aumento superior a B. La factibilidad del sistema es concerniente a las relaciones que se consideran como condición de factibilidad debido a que garantiza la factibilidad de la próxima iteración.

El nuevo punto de solución B se determina "intercambiando" la variable de entrada x1 y la variable de salida s1 en la tabla simplex para obtener:

Variables no básicas (cero) en B: (s1,x2)

Variables básicas en B: (x1, s2, s3, s4)

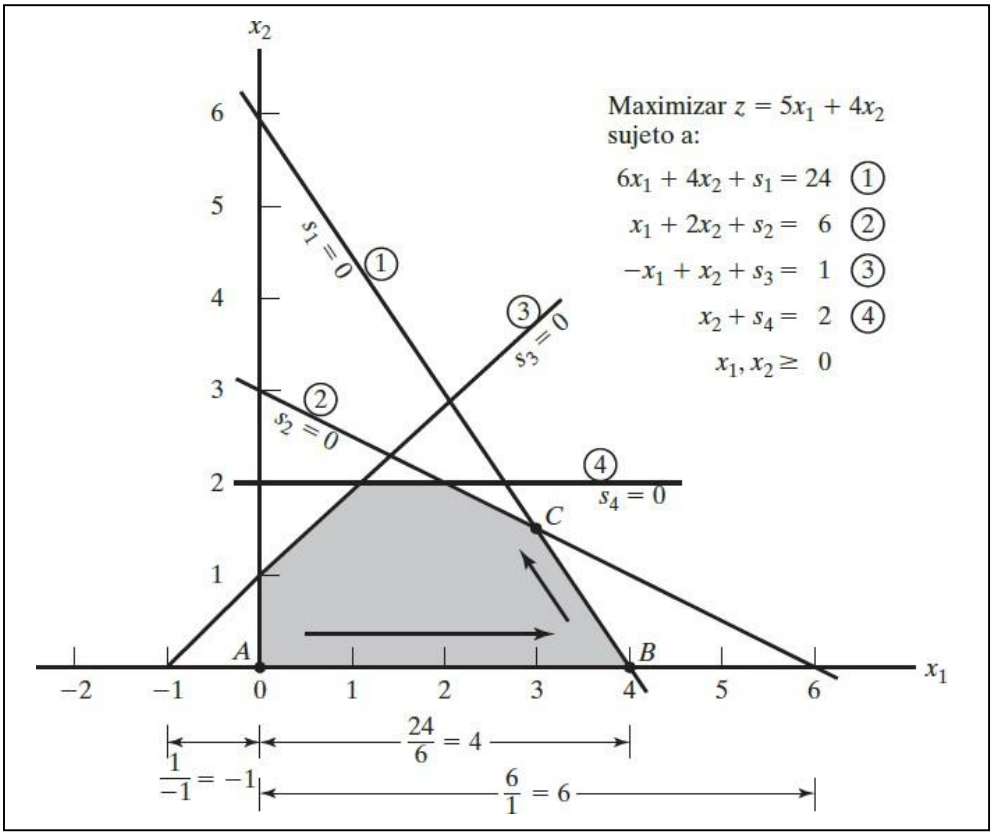


Figura 15. Interpretación gráfica de las relaciones del método simplex en el modelo

Los cálculos para las filas del método de Gauss-Jordan son la base para las iteraciones de las columnas. La definición del elemento pivote se halla mediante la intersección de la fila de salida y la columna de entrada. La primera tabla con las filas y columnas pivote se muestran en la siguiente tabla.

Tabla 2
Tabla simplex preparada para la primera iteración

Básica	z	x1	x2	s1	s2	s3	s4	Solución
z	1.0	-5.0	-4.0	0.0	0.0	0.0	0.0	0.0
s1	0.0	6.0	4.0	1.0	0.0	0.0	0.0	24.0
s2	0.0	1.0	2.0	0.0	1.0	0.0	0.0	6.0
s3	0.0	-1.0	1.0	0.0	0.0	1.0	0.0	1.0
s4	0.0	0.0	1.0	0.0	0.0	0.0	1.0	2.0

Los cálculos de Gauss-Jordan necesarios para obtener la nueva solución básica son de dos tipos:

1. Fila pivote

a. Reemplace la variable de salida en la columna básica con la variable de entrada

b. Nueva fila pivote = Fila pivote actual / Elemento pivote

2. Todas las demás filas, incluyendo z

Nueva fila = (Fila actual) – (Coeficiente de la columna pivote) x (Nueva fila pivote)

Estos cálculos se aplican a la tabla anterior como sigue:

Reemplace s1 en la columna básica con x1:

Nueva fila x1 = Fila s1 actual / 6

$$= \frac{1}{6}(0\ 6\ 4\ 1\ 0\ 0\ 0\ 24)$$

$$= (0\ 1\ \frac{2}{3}\ \frac{1}{6}\ 0\ 0\ 0\ 4)$$

Nueva fila z = Fila z actual – (-5) x Nueva fila x1

$$= (0\ 1\ \frac{2}{3}\ \frac{1}{6}\ 0\ 0\ 0\ 4)$$

$$= (1\ 0\ -\frac{2}{3}\ \frac{5}{6}\ 0\ 0\ 0\ 20)$$

Nueva fila s2 = Fila s2 actual – (1) x Nueva fila x1

$$= (0\ 1\ 2\ 0\ 1\ 0\ 0\ 6) - (1)\ x\ (0\ 1\ \frac{2}{3}\ \frac{1}{6}\ 0\ 0\ 0\ 4)$$

$$= (0\ 0\ \frac{4}{3}\ -\frac{1}{6}\ 1\ 0\ 0\ 2)$$

Nueva fila s3 = Fila s3 actual – (-1) x Nueva fila x1

$$= (0\ -1\ 1\ 0\ 0\ 1\ 0\ 1) - (-1)\ x\ (0\ 1\ \frac{2}{3}\ \frac{1}{6}\ 0\ 0\ 0\ 4)$$

$$= (0\ 0\ \frac{5}{3}\ \frac{1}{6}\ 0\ 1\ 0\ 5)$$

Nueva fila s4 = Fila s4 actual – (0) x Nueva fila x1

$$= (0\ 0\ 1\ 0\ 0\ 0\ 1\ 2) - (0)\ x\ (0\ 1\ \frac{2}{3}\ \frac{1}{6}\ 0\ 0\ 0\ 4)$$

$$= (0\ 0\ 1\ 0\ 0\ 0\ 1\ 2)$$

La nueva solución básica es (x1,s2,s3,s4) y se presenta en la Tabla 2.

Tabla 3
Tabla simplex después de la primera iteración

Básica	Z	x1	x2	s1	s2	s3	s4	Solución
z	1	0	-0.6667	0.83333	0	0	0	20
x1	0	1	0.66667	0.16667	1	0	0	4
s2	0	0	1.33333	-0.1667	1	0	0	2
s3	0	0	1.66667	0.16667	0	1	0	5
s4	0	0	1	0	0	0	1	2

El nuevo valor objetivo es z = 20, el cual es consistente con lo siguiente.

$$Nueva\ z = Anterior\ z + Nuevo\ valor\ x1\ x\ su\ coeficiente\ objetivo$$

$$= 0 + 4\ x\ 5 = 20$$

Por otra parte, z = 4 x valor de x1 + 0 x valor de s2 + 0 x valor de s3 + 0 x valor de x4 = 4 x 5 + 0 x 2 + 0 x 5 + 0 x 2 = 20.

En la última tabla, la condición de optimalidad muestra que x2 es la variable de entrada. La condición de factibilidad produce la información de la Tabla 4.

Tabla 4
Selección de variable auxiliar para hallar la relación mínima

Básica	Entrante x2	Solución	Relación
x1	2/3	4	x2 = 4/(2/3) = 6
s2	4/3	2	x2 = 2/(4/3) = 1.5 (mínima)
s3	5/3	5	x2 = 5/(5/3) = 3
s4	1	2	x2 = 2/1 = 2

Por lo tanto, s2 sale de la solución básica, y el nuevo valor de x2 es 1.5. El incremento correspondiente en z es 2/3 x2 = 2/3 x 1.5 = 1, el cual da la nueva z = 20 + 1 = 21.

Si reemplazamos s2 en la columna Básica con la x2 de entrada, se aplican las siguientes operaciones de filas de Gaus – Jordan:

Nueva fila pivote x2 = Fila s2 actual / (4/3)

Nueva fila z = Fila z actual – (-2/3) x Nueva fila x2

Nueva fila x1 = Fila x1 actual – (2/3) x Nueva fila x2

Nueva fila s3 = Fila s3 actual – (5/3) x Nueva fila x2

Nueva fila s4 = Fila s4 actual – (1) x Nueva fila x2

Estos cálculos producen la Tabla 5.

Tabla 5
Tabla con el resultado final utilizando el método simplex

Básica	Z	x1	x2	s1	s2	s3	s4	Solución
Z	1	0	0	0.75	0.5	0	0	21
x1	0	1	0	0.25	-0.5	0	0	3
x2	0	0	1	-0.125	0.75	0	0	1.5
s3	0	0	0	0.375	-1.25	1	0	2.5
s4	0	0	0	0.125	-0.75	0	1	0.5

Según la condición de optimalidad, ninguno de los coeficientes de la fila z son negativos. De ahí que la última tabla sea óptima.

3.5.3. Programación lineal entera

La PLE se define como problema de PL con el agregado en lo que respecta a la restricción que algunas variables del problema deben ser números enteros. Un PLEP o problema de programación lineal entero puro debe poseer todas sus variables enteras, mientras que una PLEM o programación lineal entera mixta puede poseer algunas variables enteras.

La programación lineal entera es aquella programación lineal en la cual necesitamos que uno, algunos o todas las soluciones sean valores enteros. Si todas las soluciones deben ser enteras, se transforma en un problema de programación lineal entera pura, caso contrario, se llama programación lineal entera mixta. (Aires, 2011).

3.6. Programación en Visual Basic para aplicaciones

Visual Basic para aplicaciones es la fusión de un entorno de programación integrado llamado Editor de Visual Basic y del lenguaje de programación Visual Basic, permitiendo diseñar y desarrollar con facilidad programas en Visual Basic. Se le añade el término “para aplicaciones” debido a que está configurado para integrarse a los programas de Microsoft Office. (Sabater, 2014).

El Editor de Visual Basic contiene todas las herramientas de programación necesarias para escribir código en Visual Basic y crear soluciones personalizadas. (Sabater, 2014).

Para escribir programas en VBA Excel se requiere el uso de su propio editor de código, en dicho editor encontraremos las opciones y funciones necesarias para crear programas personalizados.

3.6.1. Sentencias básicas

Cuando se empieza a aprender un nuevo lenguaje de programación, es casi un rito de iniciación escribir el código con el cual podamos imprimir el clásico “Hola mundo”.

Para conseguir imprimir en pantalla dicho mensaje, tenemos que hacer uso del comando MsgBox. Es importante recordar que todo código siempre debe escribirse dentro de un procedimiento Sub tal como se muestra en la Figura 16.

Al ejecutar el código de la Figura 16 se obtiene el resultado de la Figura 17.

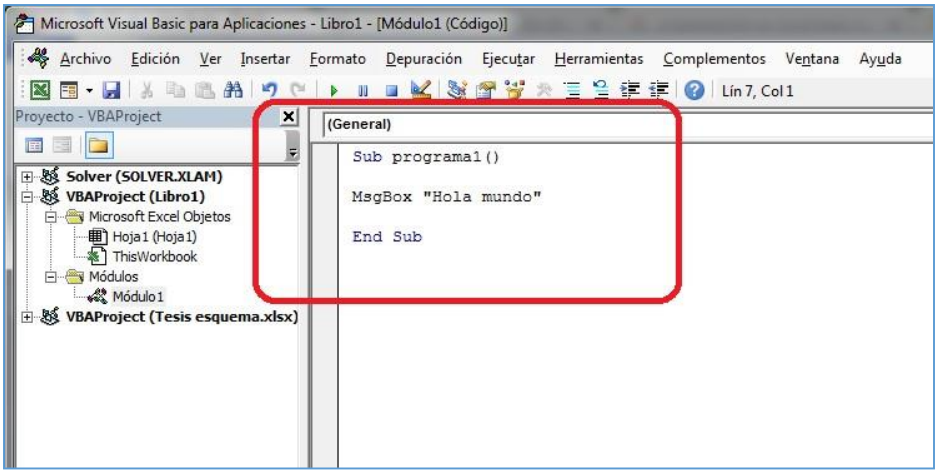


Figura 16. Código escrito en el editor de VBA para aplicaciones

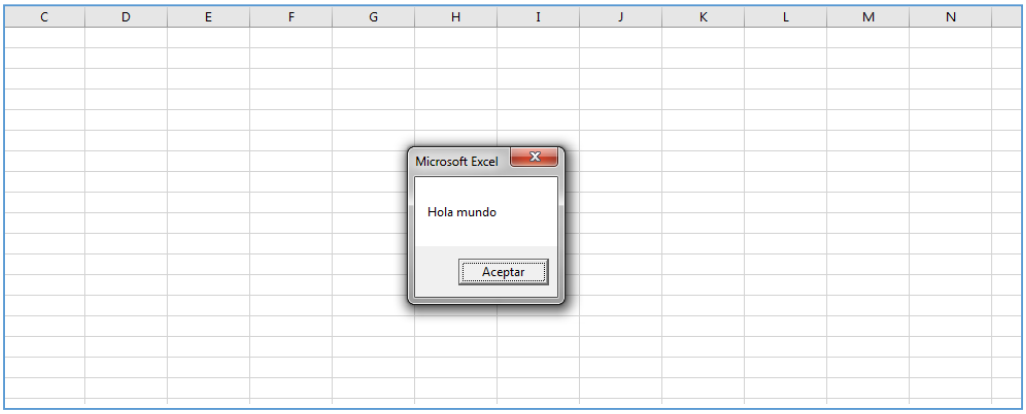


Figura 17. Mensaje "Hola mundo" impreso en pantalla al ejecutar el código

Así como se puede imprimir texto (generalmente llamados valores alfanuméricos) en pantalla usando VBA, también podemos imprimir valores numéricos.

Para mostrar mejor esta característica, utilizaremos el código para que nos imprima en pantalla el resultado de una operación aritmética (Figura 18 y Figura 19). Además, como introducción a las variables, cuyo conocimiento forma parte fundamental en el entendimiento del programa, mostraremos como se puede imprimir variables de la misma forma que las operaciones directamente (Figura 20).

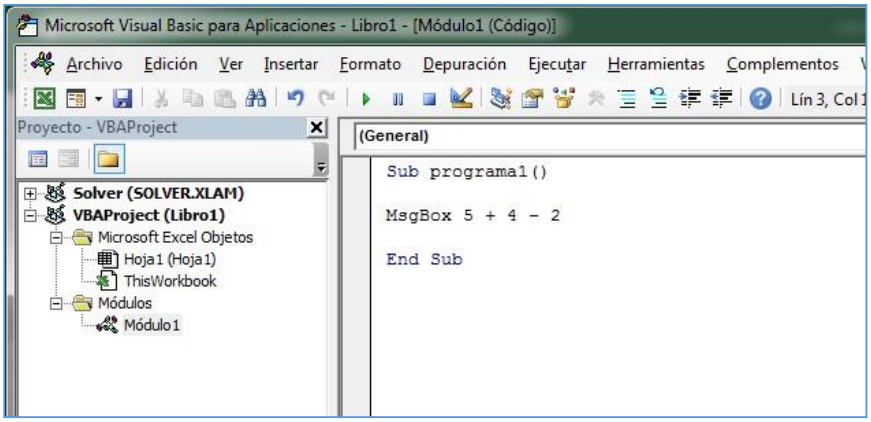


Figura 18. Código para generar el resultado de una operación en pantalla

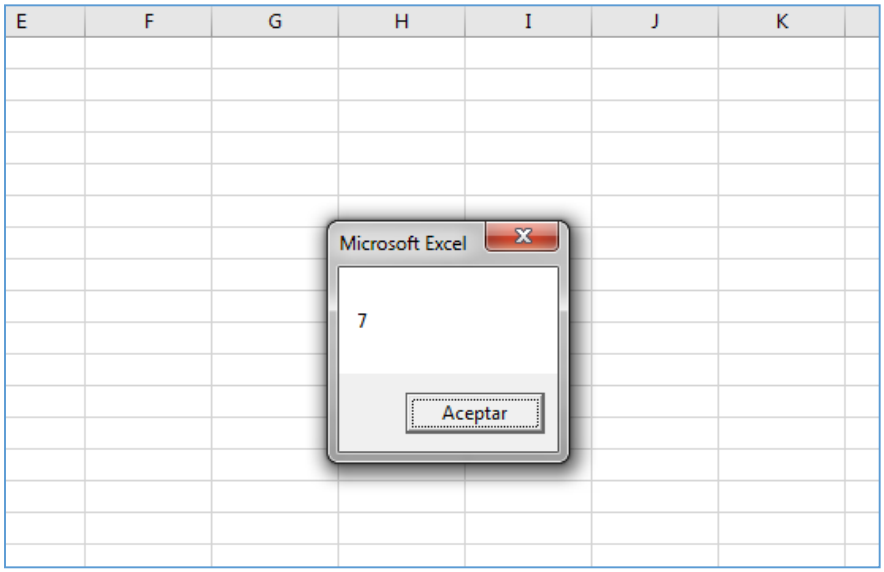


Figura 19. Mensaje con el resultado de la operación aritmética escrita en el código

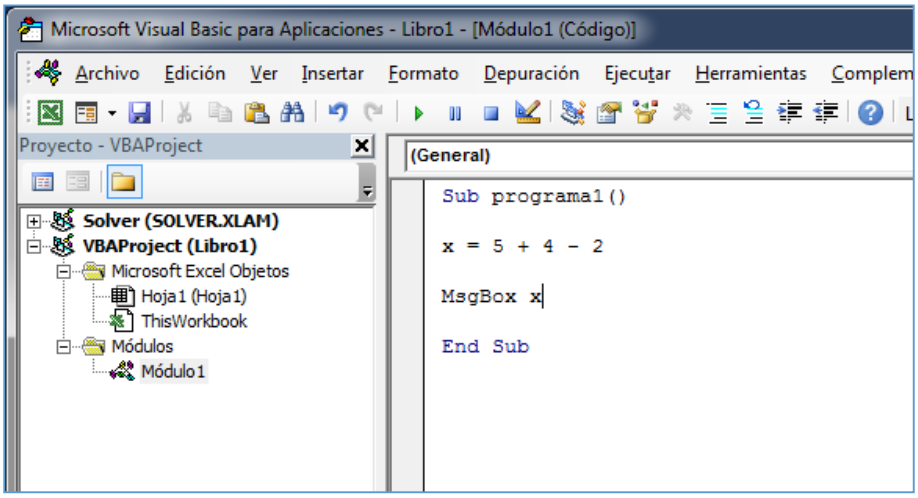


Figura 20. Uso de variable numérica en VBA

En la Figura 20, como resultado obtenemos exactamente lo mismo que la Figura 19. El uso de las variables en VBA nos permite asignar valores temporales que nos ayudarán en cierta parte el código.

3.6.1.1. Procedimientos Sub

Se ha definido al procedimiento Sub como el módulo de código que se ejecuta cuando es llamado desde algún punto del programa. (Rancel).

Todo procedimiento lleva al final de su estructura un cierre representado con la frase “End Sub”, sin embargo, se puede precipitar su conclusión con la expresión “Exit Sub”.

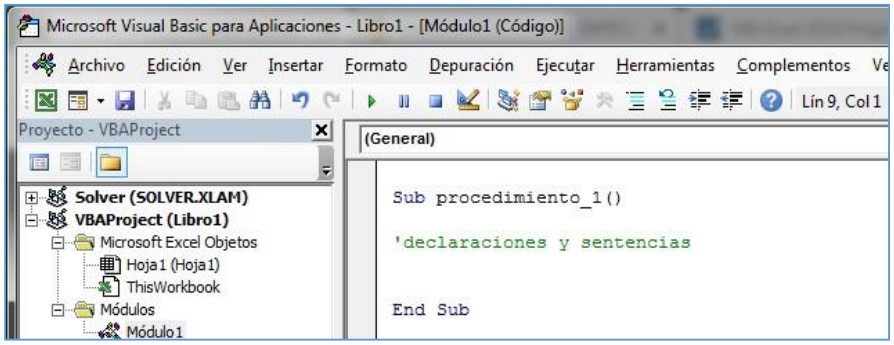


Figura 21. Estructura de un procedimiento Sub

3.6.1.2. Funciones

Una función es un conjunto de líneas de código que cuando son llamados en la ejecución de un programa devuelve un resultado, dicho resultado puede ser un valor numérico, de texto, booleano, o incluso puede ser un objeto. A continuación, se muestra la sintaxis de una función:

[Private|Public] [Static] Function nombre [(parámetros)] [As tipo]

[sentencias]

[nombre = expression]

[Exit Function]

[sentencias]

[nombre=expression]

End Function

Nombre: es la denominación del conjunto de líneas de código que constituyen la función; también se debe definir el tipo de dato que devolverá la función.

Parámetros: se denomina a los datos que son suministrados a la función. La función los utiliza en el momento que es invocada la función.

Exit Function: esta sentencia es utilizada para abortar la ejecución de una función sin la necesidad que la función culmine o que falle.

End Function: esta sentencia es colocada al final del paquete de código que representa la función. Es decir, actúa como una sentencia de cierre de la función.

Ejemplo ilustrativo

Elaboraremos un código en el cual se definirá la función que nos permita obtener el resultado de la ecuación de la energía cinética.

Como sabemos, para hallar la energía cinética de un cuerpo necesitamos conocer la masa del cuerpo y la velocidad del mismo. Podemos verlo más claro en la siguiente ecuación:

$$E = \frac{1}{2}mv^2$$

Donde:

E: energía cinética (Joule)

m: masa (Kg)

v: velocidad (m/s)

En el editor VBA de Excel crearemos la función “energía” en la cual plasmaremos la ecuación mostrada previamente. Podemos observar el código en la Figura 22.

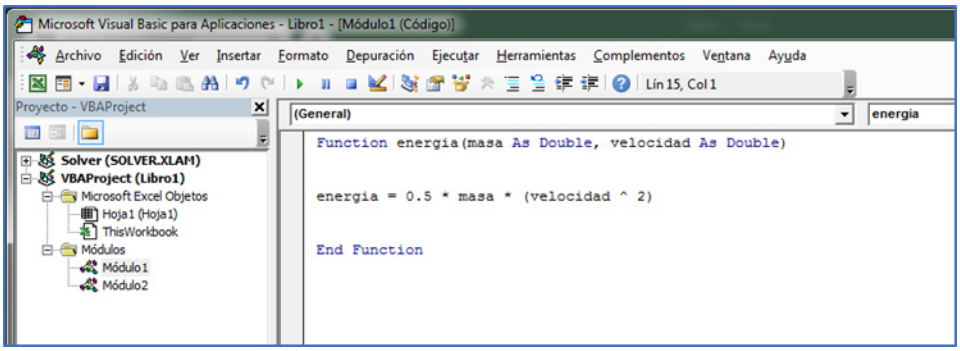


Figura 22. Código para crear la función energía

Luego, en la hoja de cálculo activa podemos hacer el llamado a la fórmula previamente creada. El llamado se hace como si se tratara de cualquier otra función predeterminada de Microsoft Excel.

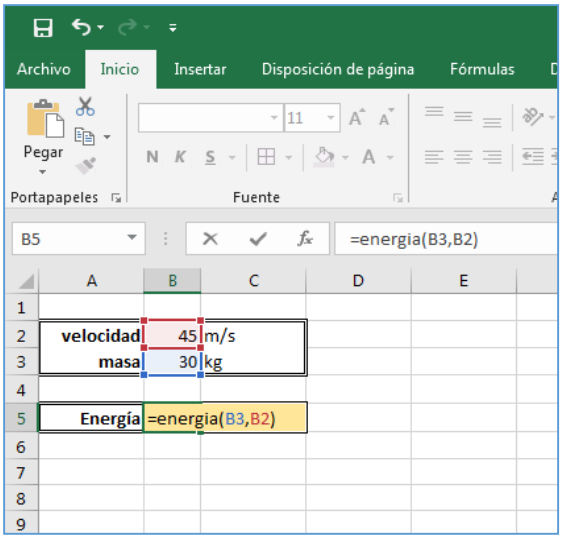


Figura 23. Llamado de la función energía desde una hoja de cálculo

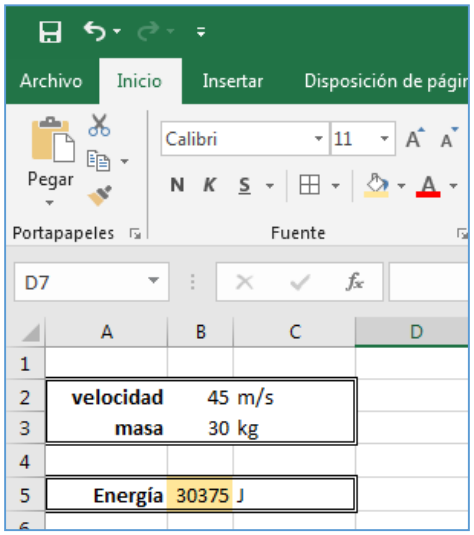


Figura 24. Hoja de cálculo mostrando el resultado en la celda sombreada

3.6.2. Constantes y variables

Se define una constante como un valor que no cambia su número durante la ejecución del programa. Las constantes nos ayudan a simplificar código debido a su practicidad en contraste con el uso de variables. Además, si todo nuestro código se relaciona con el valor de esta constante y por ejemplo, en un futuro vamos a pretender variar su valor, simplemente tendremos que cambiar el valor asignado en su declaración para que todo nuestro código, relacionado en un principio, quede actualizado correctamente.

Por ejemplo, si necesitamos cambiar un valor en un futuro solo se tendría que realizar un cambio en vez de modificar el valor en los diferentes puntos que se ha utilizado y esperar errores de no haber realizado todos los cambios necesarios.

A continuación, se muestra un ejemplo del uso de constantes en VBA.

- La constante representa el precio del oro
- El tipo de dato es decimal

- El valor numérico que asume la constante es de 1,420.25

El procedimiento en VBA será de la siguiente manera:

El código utilizado para representar el ejemplo mencionado es el siguiente:

```
Sub Calculo()

Const PrecioOro As Single = 1420.25

Dim Tonelaje As Long

Dim Ley As Single

Dim RecuperacionMetalurgica as Single

Dim ValorReserva as Single

Tonelaje = 1500

Ley = 0.5

RecuperacionMetalurgica = 0.80

ValorReserva = Tonelaje * Ley * RecuperacionMetalurgica * PrecioOro /

31.1035

MsgBox "El valor de la Reserva Mineral equivale a:" & ValorReserva

End Sub
```

Y nos brindará como resultado lo siguiente:

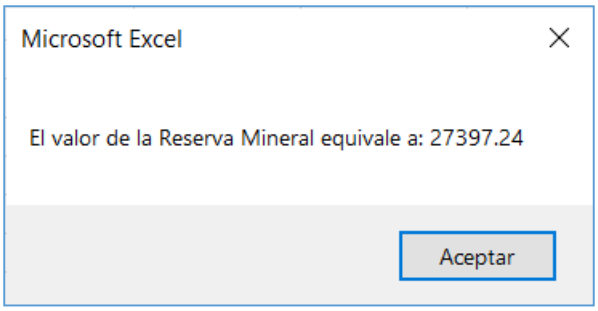


Figura 25. Resultado del ejercicio de la sección 3.6.2.

Una variable en programación es un espacio en memoria que se reserva para poder almacenar cualquier tipo de dato. Estas entidades son usadas para guardar valores que podemos usar durante la ejecución del programa.

En este lenguaje se tienen distintos tipos de variables que almacenan diversos tipos de datos, desde los que almacenan valores enteros hasta los que pueden almacenar valores booleanos. A continuación, se presenta una breve descripción de cada tipo:

Variable entera

Este tipo de variable puede almacenar número enteros. La sentencia utilizada para declararla es Integer.

Dim h As Integer

h = 10

Tal como se muestra, en la primera línea de código se declara el nombre y el tipo de variable. Luego, en la segunda línea, se le asigna un valor a dicha variable.

Variable decimal

Esta variable es capaz de almacenar número con parte decimal larga. Es decir, puede guardar valores mucho más precisos que el tipo anterior. La sentencia utilizada para declarar este tipo de variable es Double.

Dim l As Double

l = 2.2685

Este tipo de variable es capaz de almacenar números enteros también, sin embargo, no es recomendable realizar ello debido a que se estaría desaprovechando recursos del ordenador.

Variable texto

Para guardar información alfanumérica se usan variables de texto o String. Todo valor para este tipo de variable debe ser encerrado con unas comillas.

Dim Libro as String

Libro = "Programación en Excel"

Variables de tipo lógico

Una variable de tipo lógico es aquella que puede almacenar solamente dos valores: falso o verdadero. La sentencia clave para definir estas variables es Boolean.

Dim continuar As Boolean

continuar = True

La primera línea declara la variable "boleana" y en la segunda le asignamos un valor.

3.6.3. Condicional "If"

Uno de los elementos más frecuentes en VBA es usar la instrucción IF. Esta estructura nos permite dotar a nuestro código de un nivel medio de complejidad. La estructura de esta condicional es la siguiente:

If condición Then

[Sentencias a ejecutar]

[ElseIf condición2 Then]

[Sentencias a ejecutar]

[Else]

[Sentencias a ejecutar]

End if

En la estructura If, debe cumplirse la condición para que ejecute las sentencias, si no cumple con la condición el programa pasa a evaluar la condición del ElseIf, si esta condición se cumple, ejecuta las sentencias que están debajo. Finalmente, si no cumple ninguna de las condiciones anteriores, ejecuta las condiciones que se encuentran después de la instrucción Else. La condicional siempre se debe cerrar con la expresión End If.

Ejemplo Ilustrativo

Para mostrar el uso de la condicional If, presentamos un ejemplo práctico en el cual introduciremos un número entero y el programa nos dirá si este número es mayor que 50 o no.

Para llevar a cabo este ejercicio necesitamos emplear una instrucción que hasta el momento en la presente investigación no se ha mencionado. Esta instrucción se llama InputBox, la cual permite al usuario del programa ingresar datos al sistema para luego poder usarlos. Esta instrucción nos permitirá ingresarle datos numéricos o alfanuméricos al programa. Nosotros necesitamos dejar que el usuario pueda ingresar un número entero para poder evaluarlo con la condición If.

El código del programa se presenta en la Figura 25.

```
Sub procedimiento1()

    Dim valor As Integer

    valor = InputBox("Por favor, ingrese un número entero")

    If valor > 50 Then

        MsgBox "El número es mayor que 50"

    Else

        MsgBox "El número es menor que 50"

    End If

End Sub
```

Figura 26. Código VBA usando el condicional *If*

Como se puede ver en el código presentado en la Figura 26, al valor asignado por el usuario se guardará dentro de la variable entera llamada valor.

El cuadro de diálogo llamado por la sentencia InputBox aparecerá acompañado por un texto que le solicitará al usuario que digite un número entero.

Posteriormente, se observa las sentencias que conforman la estructura del If.

A continuación, en las Figuras 27, 28 y 29 se observa la prueba del programa y los resultados obtenidos al ejecutarlo.

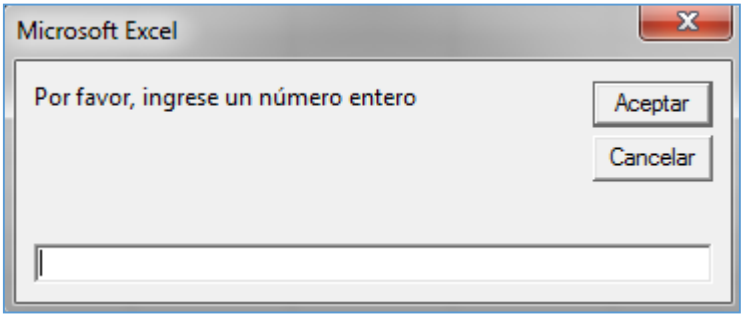


Figura 27. Ventana generada por la instrucción InputBox

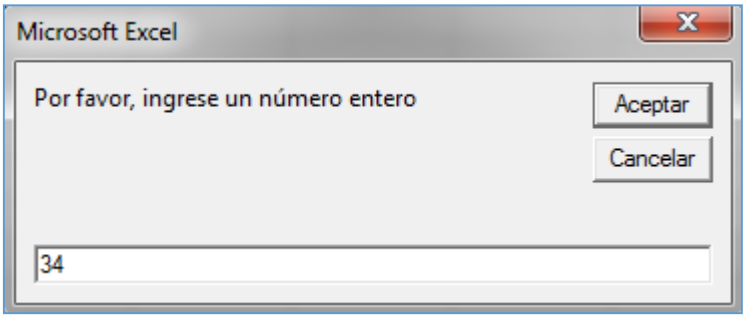


Figura 28. Ventana con el valor introducido

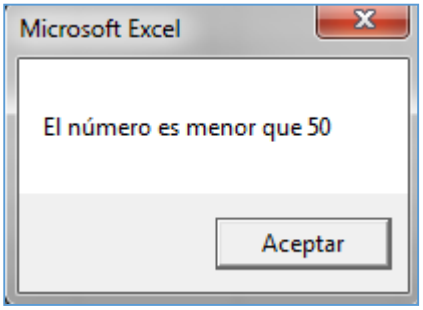


Figura 29. Ventana con el resultado del programa

El programa funciona bien, sin embargo, hay un pequeño detalle que no está correcto. Por ejemplo, si el usuario ingresa el número cincuenta, el programa dirá que el número es menor que cincuenta cuando esto no es verdad. Por lo tanto, tenemos que agregar una condición al sistema para corregir este error. El código con la corrección se muestra en la Figura 30.

```
Sub procedimiento1()

    Dim valor As Integer


    valor = InputBox("Por favor, ingrese un número entero")


    If valor > 50 Then

        MsgBox "El número es mayor que 50"

    ElseIf valor = 50 Then

        MsgBox "El número es 50"

    Else

        MsgBox "El número es menor que 50"


    End If
```

Figura 30. Ejemplo de condicional if corregido

Para corregir el error, se ha insertado una sentencia ElseIf que tomará en la evaluación si el número es igual a 50. En las Figura 31 y la Figura 32 observaremos el cumplimiento de dicha condición.

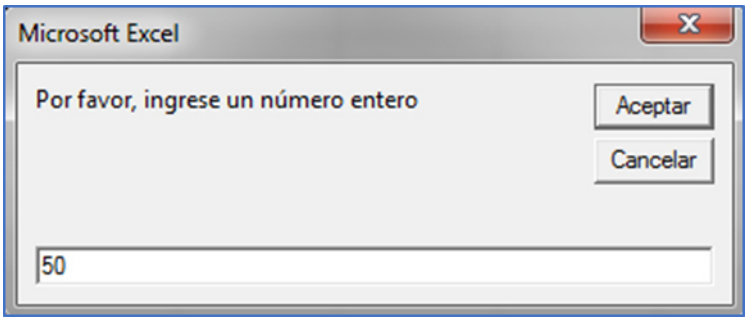


Figura 31. Comprobación de la correción del código

Figura 32. Ventana con el resultado correcto

3.6.4. Estructuras repetitivas

Estructuras de bucle de Visual Basic permiten ejecutar repetidamente una o varias líneas de código. Puede repetir las instrucciones en una estructura de bucle hasta que una condición es verdadera, hasta que una condición sea falsa, un número de veces o una vez por cada elemento especificado en una colección.

La siguiente ilustración (Figura 33) muestra una estructura de bucle que se ejecuta un conjunto de instrucciones hasta que una condición sea verdadera:

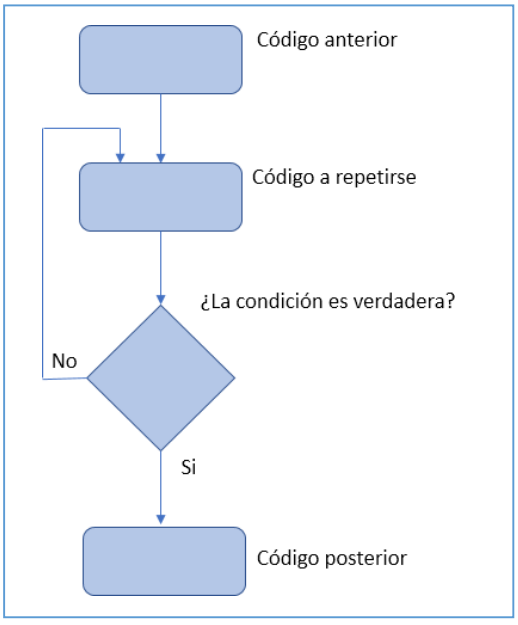


Figura 33. Diagrama de ejecución de un bucle

Bucle while

La estructura While… End While ejecuta un conjunto de instrucciones mientras la condición especificada en la instrucción While es verdadera.

Bucle do – loops

La estructura Do… Loop permite probar una condición al principio o al final de la estructura del bucle. También puede especificar si se repite un bucle mientras la condición permanezca verdadera o hasta que sea verdadera.

Bucle for

La estructura For… Next realiza el bucle un número determinado de veces. Usa una variable de control de bucle, también se denomina contador, realiza un seguimiento de las repeticiones. Especifique los valores iniciales y finales de este contador y, opcionalmente, puede especificar la cantidad por el que incrementa de una repetición a la siguiente.

Bucle for each

La estructura For Each… Next ejecuta un conjunto de instrucciones una vez por cada elemento de una colección. Especifica la variable de control de bucle, pero no es necesario determinar el valor inicial o final.

Ejemplo ilustrativo

El siguiente ejemplo consistirá en crear un código que nos permita saber cuantos alumnos han aprobado el curso y cuantos lo han reprobado (Figura 34). Este cálculo se podría realizar fácilmente usando únicamente las formulas de Excel, sin embargo, nosotros lo resolveremos empleando el bucle For Each en primera instancia y el bucle For en segunda.

Cabe resaltar que los alumnos aprobados serán aquellos que hayan conseguido una nota superior a 10, mientras que los desaprobados será aquellos que hayan tenido una nota inferior a esta.

Apellido	Nombre	Nota
Alarcón	Oscar	2
Alfageme	Jaqueline	9
Calcina	Jaime	15
Flores	Manuel	15
Garay	Javier	16
Hinostroza	Fiorella	2
Lopez	Claudio	16
Matos	Luciana	8
Perez	Pamela	13
Porles	Flor	3
Porras	Teresa	0
Robles	Luciana	20
Rosales	Hidalgo	18

Aprobados
Desaprobados

Figura 34. Relación de nota de alumnos del ejemplo

Primer método: For each

Para emplear este método es necesario crear dos variables tipo Rango (o Range). Una de estas variables nos servirá para nombrar el rango de celdas a evaluar mientras que la otra nos servirá como variable de control de flujo. También necesitaremos dos variables tipo entero, las cuales nos servirán de contadores de los alumnos aprobados y desaprobados.

Emplearemos las sentencias Set, Offset y End(xlDown). La instrucción Set nos permite asignar una celda o un rango de celdas específico a una variable tipo Range.

El comando Offset la utilizaremos para desplazarnos de una celda hacia otra. Esta instrucción siempre va acompañada de dos números entre paréntesis. Los números dentro de los paréntesis deben ser números enteros. El primer número representa al vector de las filas mientras que el segundo número representa al vector de las columnas.

A continuación, presentamos el código empleado en la Figura 35.

```
Sub contar_notas()

Dim rango1 As Range
Dim celda1 As Range
Dim aprobados As Integer
Dim desaprobados As Integer

aprobados = 0
desaprobados = 0


Set rango1 = Range(Range("B3"), Range("B3").End(xlDown))

For Each celda1 In rango1

    If celda1.Offset(0, 2) > 10 Then

        aprobados = aprobados + 1

    Else

        desaprobados = desaprobados + 1

    End If

Next celda1

Range("c18").Value = aprobados
Range("c19").Value = desaprobados


End Sub
```

Figura 35. Código usando bucle For each

Segundo método: For

Este método se diferencia del anterior debido a que el bucle no está definido en el rango de datos, por lo tanto, necesitamos contar el número de elementos que tiene el rango a evaluar. Para realizar esto, usaremos la sentencia Count como se muestra a continuación:

```
Dim k as Integer
Dim mirango1 as Range
Set mirango1 = Range(Range("a1"),range("a1").end(xldown))
For k = 1 to mirango1.count
....
Next k
```

En el ejemplo anterior se definieron dos variables, una de tipo entero y otra tipo rango. Luego, se definió el rango de datos que conforman la variable tipo rango. La variable tipo entero llamada "k" representará el número de elementos del rango de datos a evaluar en el bucle for.

A continuación, presentamos el código empleado en la Figura 36.

El resultado obtenido con ambos métodos es el mismo y se presenta en la Figura 37.

```
Sub contar_notas()

Dim rango1 As Range
Dim celda1 As Range
Dim k As Integer
Dim aprobados As Integer
Dim desaprobados As Integer

aprobados = 0
desaprobados = 0


Set rango1 = Range(Range("B3"), Range("B3").End(xlDown))

Set celda1 = Range("B3")

For k = 1 To rango1.Count

    If celda1.Offset(k - 1, 2) > 10 Then

        aprobados = aprobados + 1

    Else

        desaprobados = desaprobados + 1

    End If

Next k

Range("c18").Value = aprobados
Range("c19").Value = desaprobados


End Sub
```

Figura 36. Código usando el bucle For

	A	B	C	D	E
1					
2		**Apellido**	**Nombre**	**Nota**	
3		Alarcón	Oscar	2	
4		Alfageme	Jaqueline	9	
5		Calcina	Jaime	15	
6		Flores	Manuel	15	
7		Garay	Javier	16	
8		Hinostroza	Fiorella	2	
9		Lopez	Claudio	16	
10		Matos	Luciana	8	
11		Perez	Pamela	13	
12		Porles	Flor	3	
13		Porras	Teresa	0	
14		Robles	Luciana	20	
15		Rosales	Hidalgo	18	
16					
17					
18		**Aprobados**	7		
19		**Desaprobados**	6		

Figura 37. Resultado del ejercicio presentado en la sección 3.6.4.

3.7. Solver

Solver es un complemento incorporado dentro de Microsoft Excel cuyo objetivo principal es servir de asistente a los usuarios relacionados con problemas de optimización. Esta herramienta también brinda la posibilidad de obtener raíces aproximadas de funciones no lineales muy comunes en la ingeniería.

Diseño del modelo

Para usar el programa se debe diseñar un modelo del problema en una hoja de cálculo en un libro de Excel.

En dicho modelo se debe especificar todos los parámetros necesarios para obtener una solución óptima. El modelo se presenta en la Figura 38.

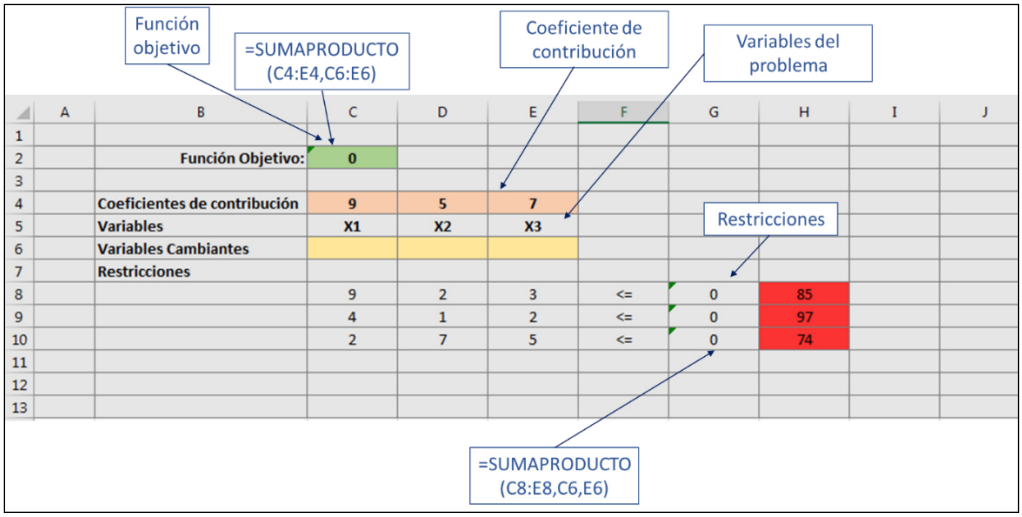


Figura 38. Modelo en hoja de cálculo de Excel

Una vez se tiene el modelo se procede a abrir Solver. Luego se definen los parámetros del solver como se detalla en el gráfico; de acuerdo al diseño elaborado en la hoja de cálculo (Figura 39). Donde la función a maximizar o minimizar y su fórmula es la suma producto de la fila de los coeficientes de contribución multiplicado por la fila de las variables cambiantes.

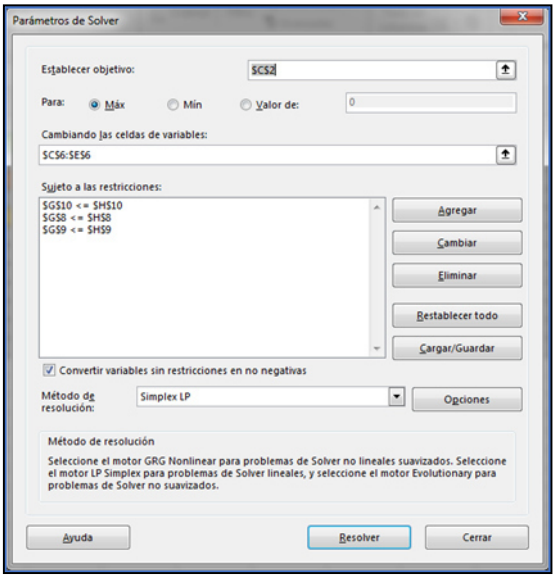


Figura 39. Parámetros a configurar en Solver

En la zona de restricciones se coloca cada una de las restricciones mencionadas anteriormente. Luego del seteo de todos los parámetros procedemos a resolver el modelo utilizando el Solver obteniendo un cuadro emergente con el resultado del modelo (Figura 40). En el gráfico se detalla que el programa ha hallado una solución que cumple todas las condiciones impuestas al sistema. De no haberse encontrado solución factible alguna, habría generado un mensaje de error.

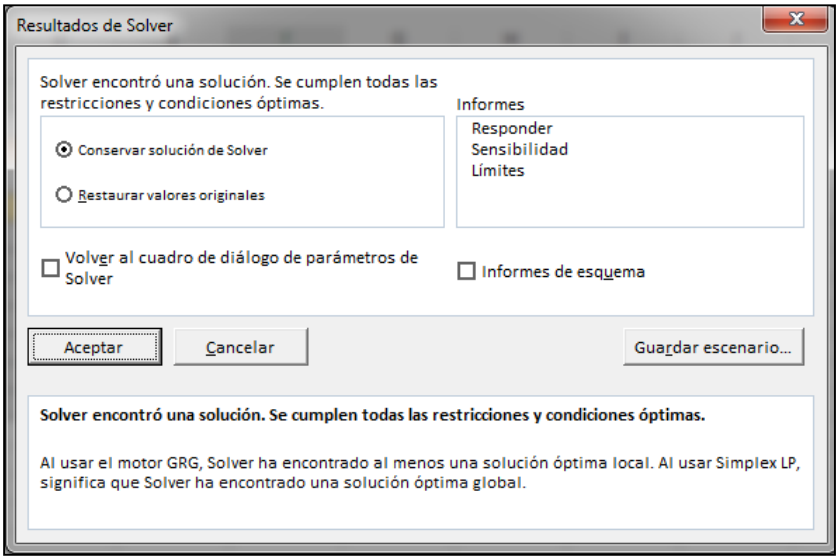


Figura 40. Cuadro emergente generado por el programa al resolver el problema

Los resultados generados por el Solver se muestran en la Figura 41.

	A	B	C	D	E	F	G	H	I	J
1										
2		**Función Objetivo:**	**135.871795**							
3										
4		**Coeficientes de contribución**	**9**	**5**	**7**					
5		**Variables**	**X1**	**X2**	**X3**					
6		**Variables Cambiantes**	5.20512821	0	12.7179487					
7		**Restricciones**								
8			9	2	3	<=	85	85		
9			4	1	2	<=	46.2564103	97		
10			2	7	5	<=	74	74		
11										
12										
13										

*Figura 41.*Resultado después de utilizar Solver

Donde el valor máximo de la función objetivo fue de 135.87 y los valores óptimo de las variables fueron de X1 = 5.205; X2 = 0; X3 = 12.718. En las restricciones se obtuvieron los siguientes resultados: la primera restricción = 85 que es el límite de la misma, la segunda restricción = 46.25 que es menor a 97, la tercera restricción = 74 que es la cota superior de la misma.

Capítulo IV

Análisis de resultados

En el presente capítulo se dará a conocer los resultados de la presente investigación. También se mostrará brevemente los pasos seguidos para obtener los resultados. Cabe destacar el programa utilizado para poder generar los datos que nos servirán como datos de entrada en la resolución del problema es el MineroSuite de la empresa Minero INC.

4.1. Datos de entrada

Los datos de entrada que utilizaremos serán los siguientes:

- Topografía del proyecto, curvas de nivel a medio banco.

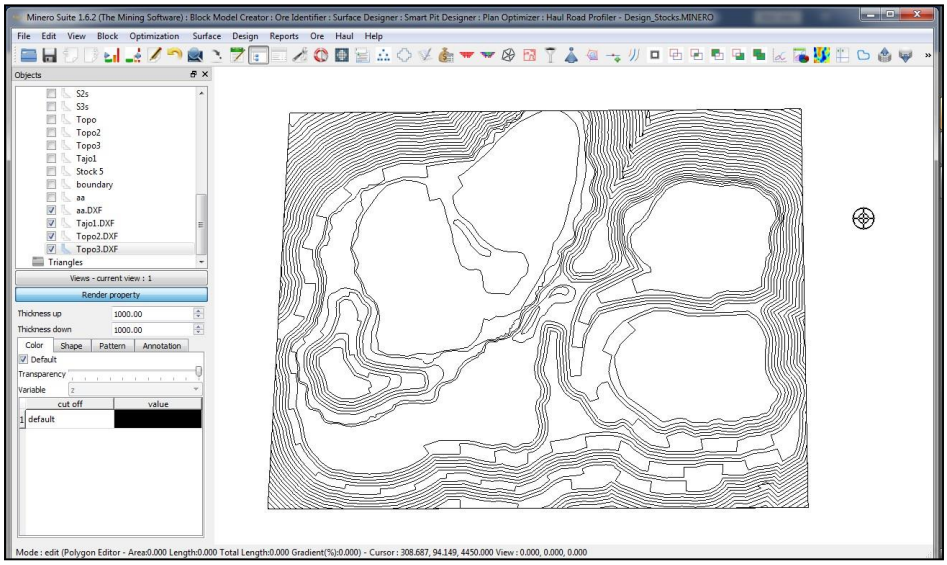


Figura 42. Topografía inicial del proyecto

- Modelo de bloques para los stockpiles con las variables más importantes del modelo como son plata, cobre y oro.

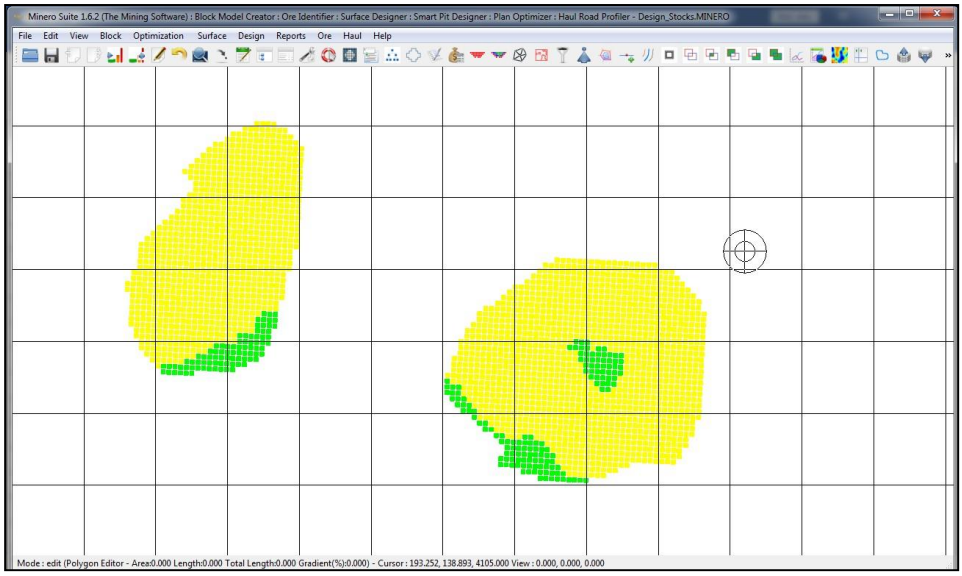


Figura 43. Vista del modelo de bloques cargado en el software Minero Suite

- Sólidos que representan los sólidos de cada stockpile.

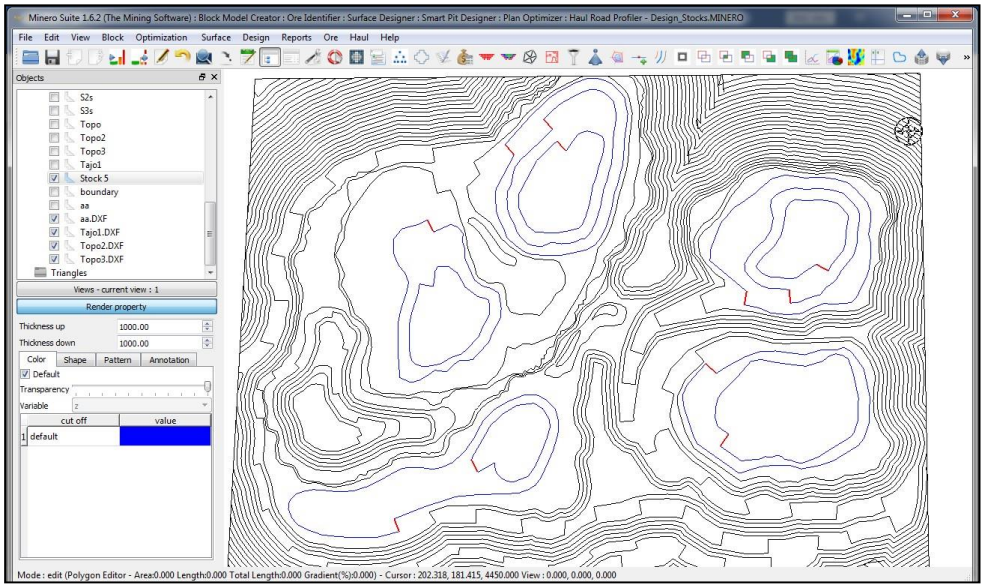


Figura 44. Lineas medias de los stockpiles con la topografía

- Parámetros económicos

  Costo de remanejo = 2.00 $/ton

  Costo de procesamiento = 5.50 $/ton

  Gastos generales = 2.00 $/ton

  Precio Ag = 14.50 $/oz

  Factor = 0.65

  Recuperación metalúrgica = 25%

Con los parámetros económicos podemos hallar la ley de corte empleando la siguiente fórmula:

$$Ley\ de\ corte = \frac{(CR + CP + GG)}{P\ x\ RM\ x\ F}$$

Donde:

CR: Costo de remanejo

CP: Costo de procesamiento

GG: Gastos generales

P: Precio de la plata

RM: Recuperación metalúrgica

F: Factor

Reemplazando los valores en la fórmula:

$$Lc = \frac{(2.00 + 5.50 + 2.00)}{14.50\ x\ 0.25\ x\ 0.65} = 4.03\,\frac{oz}{ton} = 4\ oz/ton$$

Esta ley de corte la utilizaremos en la sección siguiente para poder clasificar el material en mineral y desmonte respectivamente.

4.2. Clasificación de material

En esta sección, diferenciaremos el material en 4 grupos: ley alta, ley media, ley baja y desmonte. El rango de ley para cada grupo se presenta en la Tabla 6.

En el programa, para poder llevar esta clasificación al modelo de bloques, se creó una variable llamada "CLASSIFICATION". En esta variable se colocará el tipo de material dependiendo de su ley.

En el programa MineroSuite, existe un módulo llamado Compute variable. Este módulo nos permite escribir scripts con los que se puede realizar operaciones con las variables del modelo de bloques.

Tabla 6
Clasificación de material según su ley de plata

Clasificación	Rango de ley
Desmonte	0 - 4 oz
Ley baja	4 - 6 oz
Ley media	6 - 8 oz
Ley alta	> 8 oz

El módulo Compute variable de MineroSuite permite escribir el código en el lenguaje JavaScript. Se pueden ejecutar estructuras condicionales como el if y bucles como el for. Para el presente problema se utilizó el if para condicionar el rango de leyes de plata como la clasificación. El código se presenta en la Figura 45.

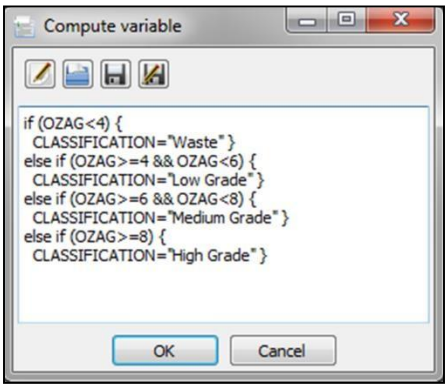


Figura 45. Código para clasificar el material dentro del modelo

Después de ejecutar el código, codificamos el modelo de bloques para verlo, en la Figura 46, según la siguiente leyenda:

Desmonte: azul

Ley baja: verde

Ley media: amarillo

Ley alta: rojo

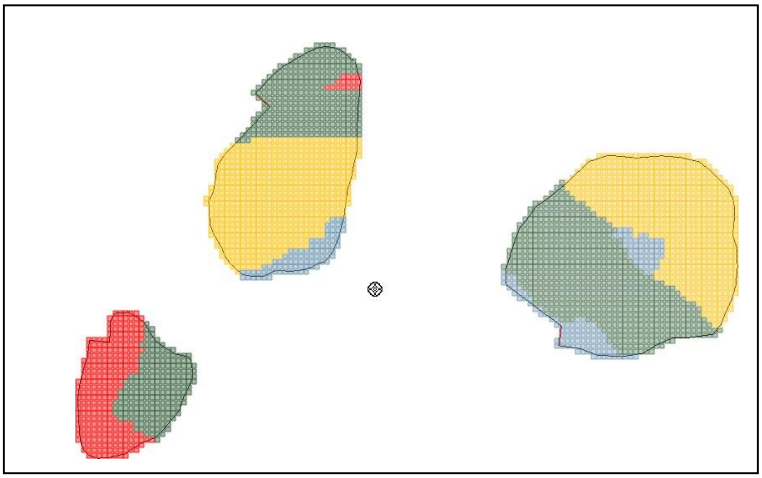


Figura 46. Modelo de bloques de los stocks clasificado por material

El anterior procedimiento se ejecutó para cada stock. Se obtuvo una tabla de tonelajes y ley de plata por cada clasificación que se muestra en la Tabla 7. En total se generaron 8 tablas, una por cada stockpile.

Tabla 7
Clasificación de material del stock 1 con tonelaje y ley

Stock 1				
Clasificación	**Rango**	**Tonelaje (t)**	**OZAG (oz/t)**	**Cu (%)**
Desmonte	**0 - 4 oz**	184,218	3.153	0.229
Ley baja	**4 - 6 oz**	328,511	4.731	0.192
Ley media	**6 - 8 oz**	60,327	6.521	0.254
Ley alta	**> 8 oz**	-	0	0
Total Mineral		**388,838**	**5.009**	**0.202**

Esta información la usamos para realizar un primer análisis económico.

En la Tabla 8 vemos un cuadro resumen de todos los stocks, notamos que el stock 3 no debe ser remanejado. Este stock tiene muy poco mineral para ser recuperado y posee mucho desmonte que se debe extraer.

Tabla 8
Tabla resumen de los stocks

Stock	Mineral		Desmonte	% Mineral
	Tonelaje (t)	OZAG (oz/t)	Tonelaje (t)	
1	405,645	5.22	59,410	69%
2	100,712	5.63	47,411	41%
4	77,150	10.03	-	100%
5	38,932	4.53	229	98%
6	41,027	6.24	918	94%
7	69,463	7.58	-	100%
8	122,609	7.73	-	100%

En los procedimientos siguientes no se considerará el Stock 3. A partir de este punto de la investigación, solo contaremos con siete stocks.

4.3. Generación y dimensionamiento de polígonos

Ahora que se tiene clasificado el modelo de bloques podemos pasar a generar los polígonos de minado según el tipo de material que posee.

La necesidad de generar polígonos de minado nace del objetivo principal de la presente investigación, la cual es, generar una secuencia de extracción. Por lo tanto, necesitamos definir una unidad de secuenciamiento.

La unidad de secuenciamiento será la unidad desde la cual se secuenciarán los stocks. Esta unidad de secuenciamiento debe dimensionarse de acuerdo al horizonte que se plantee programar. Por lo tanto, mientras más grande sea nuestro horizonte (años), la unidad de secuenciamiento debe ser grande también (100,000 toneladas deben tener cada unidad de secuenciamiento por dar un ejemplo).

Para la presente investigación, el tamaño del polígono será de 2,500 toneladas. Este valor fue impuesto por el cliente al cual se le estaba brindando la asesoría, por lo tanto, este dato fue un valor de entrada en lugar de un valor calculado.

Sin embargo, primero debemos generar los polígonos de minado según la clasificación en el modelo de bloques.

Para realizar este paso de la investigación utilizaremos dos cosas:

1) La variable Classification creada previamente en el modelo de bloques
2) Los módulos Compute variable y Mine Polygon Design del software Minero Suite

Primero, creamos las variables binarias LB, LM, LA, DE. Estas variables tomarán el valor uno, solamente cuando el bloque pertenezca al tipo de clasificación asignada. Las asignaciones son las siguientes:

- LB: ley baja
- LM: ley media
- LA: ley alta
- DE: desmonte

Para ilustrar como se introduce esta idea al programa utilizaremos el módulo Compute variable de MineroSuite y escribiremos el código de la Figura 47.

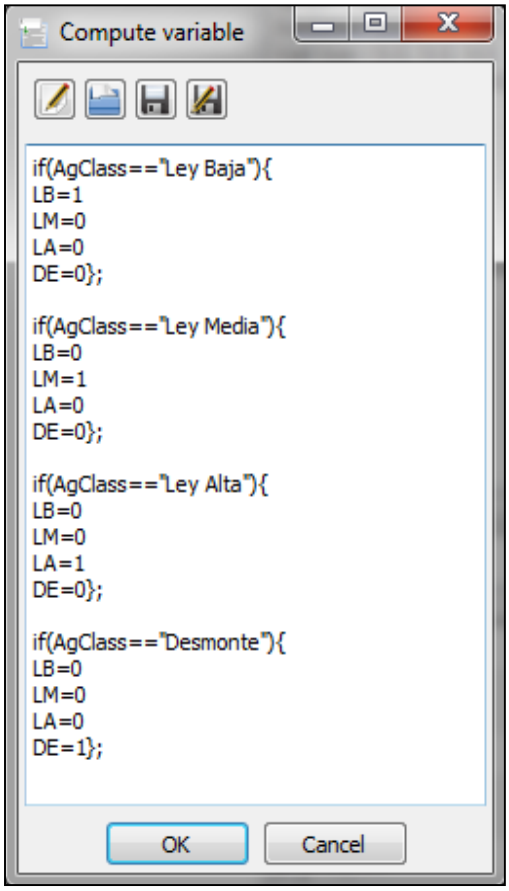


Figura 47. Código empleado para configurar las variables LB, LM, LA, DE

Luego, tenemos que utilizar el módulo Mine polygon design de MineroSuite. Este módulo nos permite utilizar las variables creadas y configuradas anteriormente con el módulo Compute variable para poder generar polígonos según su clasificación de material analizada en la sección anterior. Además, este módulo genera los polígonos teniendo en cuenta un tamaño mínimo del lado de polígono que el usuario configura. Esto nos permite añadir al diseño de polígonos la restricción del ancho del equipo de carguío, por lo tanto, el programa no generará polígonos en los cuales el equipo de carguío no podrá minar. Mine polygon design se presenta en la Figura 48.

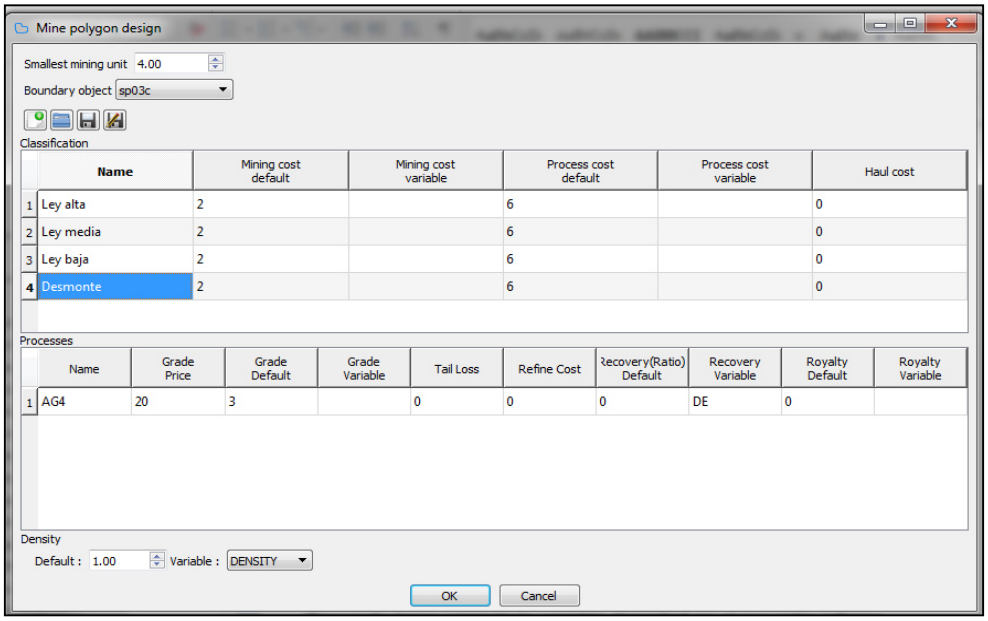


Figura 48. Ventana del modulo Mine polygon design de MineroSuite

Luego, asignamos la variable densidad del modelo al módulo. Finalmente, para ejecutar el módulo, tenemos que hacer click en OK en la ventana del módulo. Los polígonos según su clasificación dentro de los stocks quedarían como se muestra en la Figura 49.

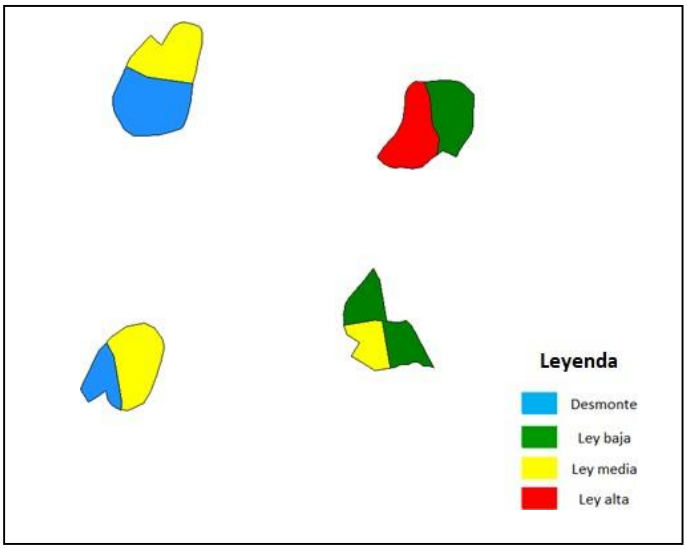


Figura 49. Polígonos generados según la clasificación del material

En la sección 3.4.2 se explicó cómo se puede segmentar un polígono en varios polígonos con un tamaño determinado usando MineroSuite. El mismo procedimiento seguimos para segmentar los polígonos de los stocks generados con el *Mine polygon design,* estos polígonos deben de ser de 2,500 toneladas la cual será nuestra unidad de planificación. El resultado los observaremos en la Figura 50.

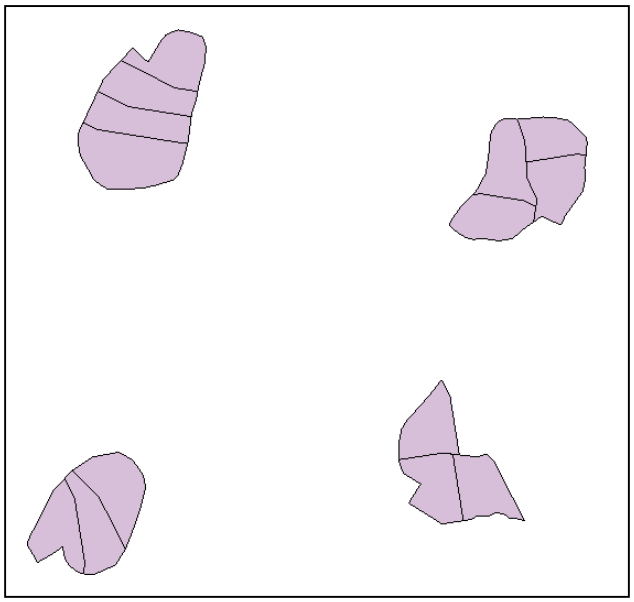


Figura 50. Polígonos segmentados equivalente a la unidad de planificación

Para poder secuenciar los polígonos con el fin de generar el plan de extracción necesitamos crear el inventario de cada stock. Para ello, utilizamos el módulo *Grade tonnage*, el cual es el módulo de MineroSuite encargado de generar los reportes de tonelaje, ley y atributos de los modelos dentro del sistema (Figura 51).

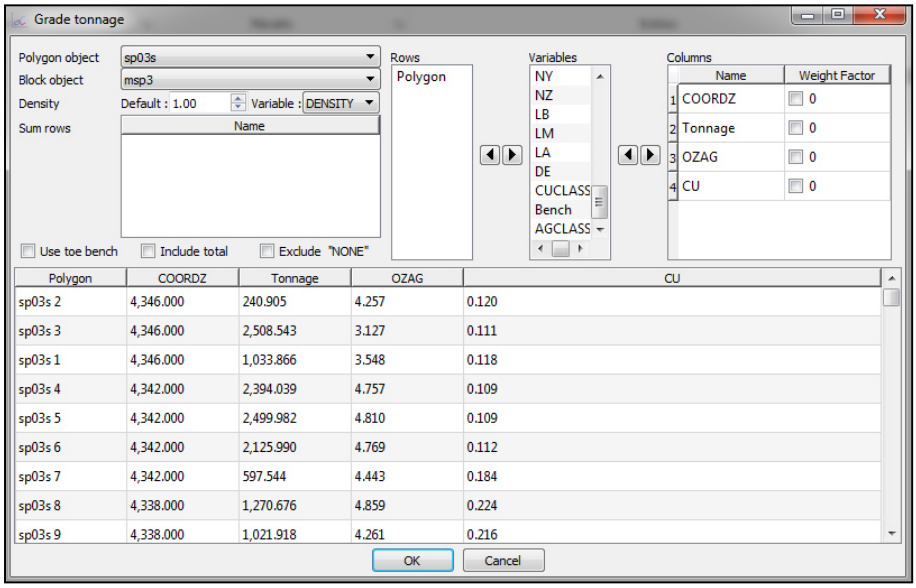
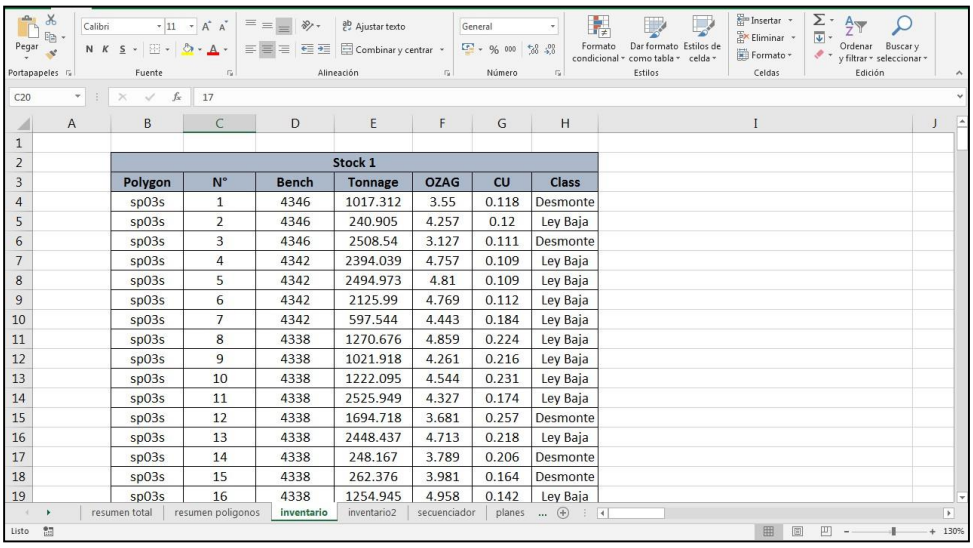


Figura 51. Interfaz de usuario del modulo *Grade tonnage*

Luego, estos reportes los guardamos y copiamos dentro de una hoja de Microsoft Excel del Libro que contiene el secuenciador. Específicamente se guardará en la hoja "Inventario".



		Stock 1					
	Polygon	N°	Bench	Tonnage	OZAG	CU	Class
	sp03s	1	4346	1017.312	3.55	0.118	Desmonte
	sp03s	2	4346	240.905	4.257	0.12	Ley Baja
	sp03s	3	4346	2508.54	3.127	0.111	Desmonte
	sp03s	4	4342	2394.039	4.757	0.109	Ley Baja
	sp03s	5	4342	2494.973	4.81	0.109	Ley Baja
	sp03s	6	4342	2125.99	4.769	0.112	Ley Baja
	sp03s	7	4342	597.544	4.443	0.184	Ley Baja
	sp03s	8	4338	1270.676	4.859	0.224	Ley Baja
	sp03s	9	4338	1021.918	4.261	0.216	Ley Baja
	sp03s	10	4338	1222.095	4.544	0.231	Ley Baja
	sp03s	11	4338	2525.949	4.327	0.174	Ley Baja
	sp03s	12	4338	1694.718	3.681	0.257	Desmonte
	sp03s	13	4338	2448.437	4.713	0.218	Ley Baja
	sp03s	14	4338	248.167	3.789	0.206	Desmonte
	sp03s	15	4338	262.376	3.981	0.164	Desmonte
	sp03s	16	4338	1254.945	4.958	0.142	Ley Baja

Figura 52. Inventario del Stock 1 almacenado en la hoja de cálculo Inventario

Realizamos la misma acción para cada stock.

4.4. Algoritmo de programación lineal entera mixta

Desde el comienzo de la presente investigación, se ha planteado la hipótesis de poder conseguir un secuenciamiento a partir de un algoritmo de programación lineal entera mixta. Por consiguiente, en este punto explicaremos el funcionamiento del mismo. Para hacer esta explicación más sencilla, he preparado un ejercicio similar al problema del presente estudio, pero con una menor cantidad de variables y restricciones.

Iniciamos con la premisa que tenemos tres stocks divididos en cuatro partes cada uno (Figura 53).

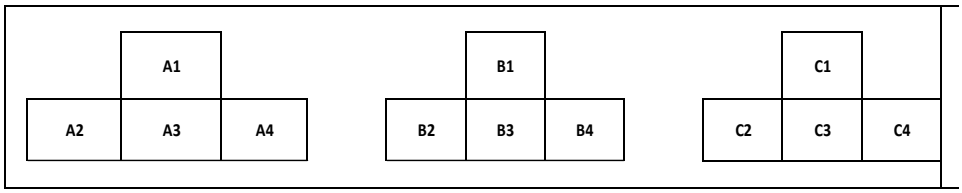


Figura 53. Diagrama de tres stocks divididos en cuatro partes

Cada parte de cada stock tiene asignado un valor económico, el cual depende de su ley de oro principalmente. El valor económico de cada bloque se presenta en la Figura 54.

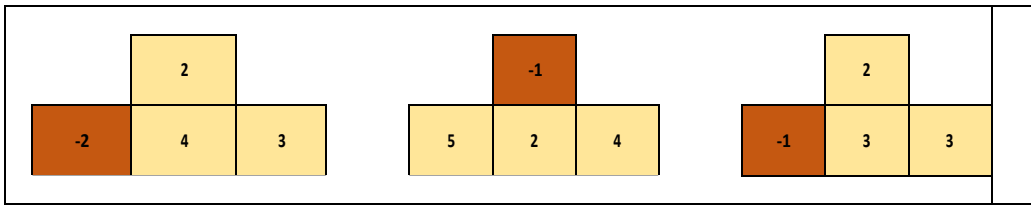


Figura 54. Esquema con los valores de cada bloque de los stocks

Otro dato que debemos tener en cuenta en este ejemplo es que cada bloque, el cual representa la unidad de planificación, tendrá un tonelaje de cincuenta toneladas. Es decir, cada stock tendrá doscientas toneladas de material.

Se buscará planificar la extracción de los stocks en tres periodos. En cada periodo, el tonelaje que debe extraerse de todos los stocks debe ser de doscientas toneladas. Debe maximizarse el valor total en cada periodo.

Ahora, configuramos la decisión de extraer cada uno de los bloques en cada uno de los periodos como una variable binaria, tal como se muestra en la Tabla 9.

Como se puede ver, el número de variables es de 36, el cual resulta de la multiplicación del número de bloques total (12) y el número de periodos (3).

Para comenzar a configurar el algoritmo de programación lineal entera mixta, formulamos la función objetivo a continuación.

$$Max\ z = \sum_{t=1}^{T} \sum_{n=1}^{K} b_n^t \,.\, V_n$$

Donde:

b_n^t: variable binaria del bloque n en el periodo t

V_n: valor del bloque n

K: número total de bloques

T: número total de periodos

Luego que hemos definido la función objetivo, tenemos que plantear las restricciones aplicadas al sistema.

$$\sum_{i \in SP} \sum_{n=1}^{K} b_{n-1}^{t,i} \leq \sum_{i \in SP} \sum_{n=1}^{K} b_n^{t,i} \ldots (1)$$

$$\sum_{n=1}^{K} b_n^t \,.\, T_n = T \ldots (2)$$

$$\sum_{n=1}^{K} b_n^t \,.\, V_n \leq V \ldots (3)$$

Donde:

SP: conjunto de stockpiles en el sistema

T_n: tonelaje del bloque n

T: $tonelaje\ objetivo\ en\ el\ periodo\ t$

V: $valor\ objetivo\ en\ el\ periodo\ t$

La restricción (1) se refiere al orden en el cual se extraerán los bloques del stockpile, por ejemplo, no se puede extraer el bloque A2, A3 o A4 del stock 1 sin antes haber extraído el bloque A1. Lo mismo con los otros dos stocks restantes.

La restricción (2) fija un tonelaje para cada periodo que se desea planificar, es decir, la suma del tonelaje de los bloques seleccionados para ser extraídos en cada periodo debe ser iguales al tonelaje objetivo.

La restricción (3) fija un valor económico máximo a cada periodo. De esta manera, se asegura que ningún periodo llegue a ser antieconómico y por lo tanto que el número de periodos se reduzca.

Ahora, introducimos la función objetivo y las restricciones dentro del panel de Solver de Microsoft Excel (Figura 55). Al presionar Ok, el programa dará el valor óptimo a las celdas donde tenemos guardadas variables binarias, es decir, a través del valor que introduzca en dichas celdas (0 o 1) determinará el periodo en el cual se debe extraer cada bloque de cada stock. Los resultados se muestran en la Figura 56.

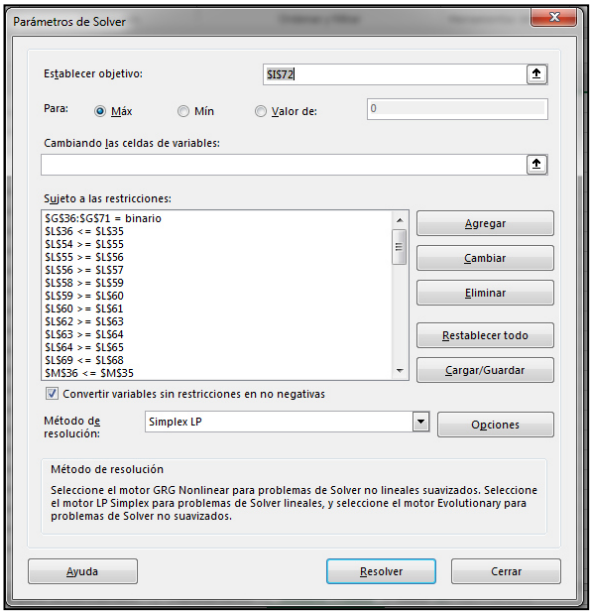


Figura 55. Ventana de Solver con los parámetros del ejemplo de la sección 4.4.

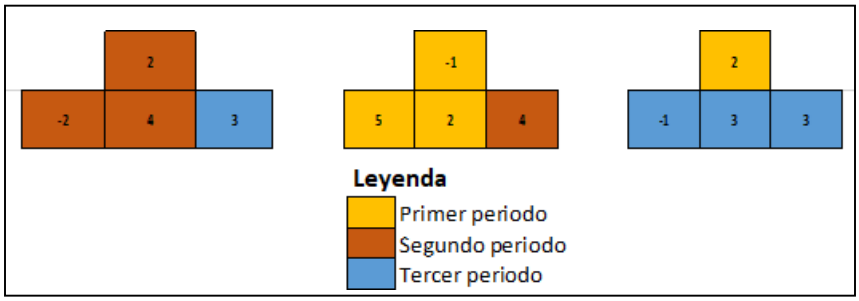


Figura 56. Secuencia de extracción del ejemplo de la sección 4.4.

Como se puede ver en la Figura 56, todos los periodos tienen el mismo tonelaje. Cada periodo tiene cuatro bloques, los cuales suman doscientas toneladas. Además, podemos comprobar que el valor en cada periodo es el mismo. Este valor total se puede calcular sumando el producto de cada bloque del periodo por su tonelaje.

Para el presente caso, el valor del periodo 1se calcularía de la siguiente manera:

V1 = -1 x 50 + 5 x 50 + 2 x 50 + 2 x 50 = 400

Si realizamos el mismo procedimiento en los otros periodos, se comprobará que el valor es igual en todos los periodos.

Ahora que ya explicamos cómo se aplica el algoritmo de programación lineal mixta entera para conseguir planes de extracción a partir de la función objetivo y las restricciones podemos presentar el algoritmo que se empleó para la presente investigación.

El esquema para el presente estudio es muy parecido al de Askari Nasab presentado en el capítulo 3. Lo que lo diferencia es que este esquema presenta el caso contrario al que presenta Askari, en lugar de tener un solo origen, el cual sería el tajo en el estudio de Askari; en el presente estudio se tienen varios orígenes, siete orígenes para ser más precisos, a los cuales representa cada stock.

En resumen, tenemos siete stocks desde los cuales saldrá el material y tenemos solo dos destinos, la chancadora primaria y el botadero de desmonte. El destino del material dependerá de la clasificación de material asignada al polígono en los apartados anteriores.

Ahora, lo que hemos categorizado como orígenes de material, lo hemos desglosado anteriormente en polígonos por cada stock y lo hemos guardado dentro de la hoja de cálculo “Inventario” del programa.

Si observamos este inventario, tenemos 594 polígonos de minado sumarizando los siete stocks. Por lo tanto, si planteamos el algoritmo mostrado en el ejemplo anterior sin hacer algún filtro dentro del inventario tendremos 594 variables, y esto, considerando únicamente un periodo a secuenciar. Si quisiéramos un plan de extracción de doce meses, este número de polígonos tendría que multiplicarse por el número de periodos y nos daría el siguiente número de variables.

variables = 594 x 12 = 7,128 variables

Trabajar con un número tan grande de variables es muy complicado, no a nivel estructural porque para estos tipos de cálculo iterativos es que vamos a usar el programa Visual Basic. El problema es con el tiempo de procesamiento y que el tema del presente estudio es aprovechar el complemento del Microsoft Excel llamado Solver para poder implementar sobre éste el algoritmo de programación lineal y este programa tiene como máximo cien variables a manejar.

Por lo tanto, vamos a crear un sub-inventario, un lugar donde carguemos los diez primeros polígonos de cada stock, desde el cual el programa tendrá que encontrar la combinación para maximizar la ley de plata y mantener la ley de cobre dentro de los límites máximos permisibles.

Además, cabe resaltar que se ejecutará el programa de secuenciamiento una vez por carga de los setenta polígonos, lo cual no asegura que en cada corrida logremos secuenciar un periodo completo, esto dependerá de que tan bien se preste el inventario de dicha corrida para cumplir las restricciones. Podría darse el caso que varias corridas del programa conformen un periodo. Esta característica del programa en lugar de verse como una limitante o defecto, puede verse como un valor añadido debido a que le da al programa una versatilidad para ir paso a paso consiguiendo el secuenciamiento y nos permite en alguna etapa del proceso cambiar ciertas restricciones con el fin de flexibilizar el plan.

Para conseguir que estos pasos no sean tan manuales, se ha contemplado mecanismos por los cuales se pueda actualizar el inventario de polígonos. Es decir, una vez que un polígono ya se haya seleccionado a través del programa para extraerse en un periodo determinado, que desaparezca del inventario para evitar el problema de secuenciarlo dos veces o más.

Ahora que hemos mecionado todas las condiciones que se tuvieron en cuenta, presentamos el algoritmo usado en la presente investigación:

Función objetivo

$$z = \sum_{i \in SP} \sum_{n \in Ore} \sum_{n=1}^{K} b_n \,.\, T_n.\, OZAG_n$$

Restricciones

$$\sum_{i \in SP} \sum_{n=1}^{K} b^{\,i}_{n-1} \leq \sum_{i \in SP} \sum_{n=1}^{K} b^{i}_{n} \ldots (1)$$

$$\sum_{i \in SP} \sum_{n \in Ore} \sum_{n=1}^{K} b_n.\, T_n.\, OZAG_n \leq OZAG_{max} \ldots (2)$$

$$Cu_{min} \leq \sum_{i \in SP} \sum_{n \in Ore} \sum_{n=1}^{K} b_n.\, T_n.\, Cu_n \leq Cu_{max} \ldots (3)$$

$$Tore_{min} \leq \sum_{i \in SP} \sum_{n \in Ore} \sum_{n=1}^{K} b_n \,.\, T_n \leq Tore_{max} \ldots (4)$$

$$Twst_{min} \leq \sum_{i \in SP} \sum_{n \in Wst} \sum_{n=1}^{K} b_n \,.\, T_n \leq Twst_{max} \ldots (8)$$

Donde:

$\boldsymbol{b_n}$: *variable de binaria de decisión del polígono n*

$\boldsymbol{T_n}$: *tonelaje del polígono n*

$\boldsymbol{OZAG_n}$: *ley de plata del polígono n*

$\boldsymbol{i} \in \boldsymbol{SP}$: *cada stockpile del sistema*

$\boldsymbol{n} \in \boldsymbol{Ore}$: *cada polígono que es considerado mineral*

$\boldsymbol{OZAG_{max}}$: *contenido metálico de plata máximo permitido*

$\boldsymbol{Cu_n}$: *ley de cobre del polígono n*

$\boldsymbol{Cu_{min}}$: *contenido metálico de cobre mínimo permitido*

$\boldsymbol{Cu_{max}}$: *contenido metálico de cobre máximo permitido*

$\boldsymbol{Tore_{min}}$: *tonelaje de mineral mínimo requerido*

$\boldsymbol{Tore_{min}}$: *tonelaje de mineral mínimo requerido*

4.5. Evaluación económica

En esta sección del capítulo cuatro valorizaremos el plan de extracción de los stocks, consideraremos los parámetros económicos utilizados en la sección 4.1 para hallar la ley de corte. También, mostraremos la influencia del contenido de cobre en el beneficio.

En primer lugar, mostraremos una tabla con los tonelajes y leyes para cada mes generado por el programa (Tabla 9).

En esta tabla podemos observar que a partir del noveno mes la ley de cobre se incrementa por encima del límite establecido.

En el Anexo 1 se presentan los cortes generados para cada mes del secuenciamiento.

Tabla 9
Resumen de plan de extracción mensual

Mes	Desmonte (t)	Mineral (t)	Leyes		Total Material
			OZAG (oz/t)	CU (%)	(t)
1	10,392	77,377	6.82	0.13	87,769
2	8,857	73,028	6.78	0.13	81,885
3	9,327	74,658	6.71	0.12	83,985
4	14,050	75,440	6.62	0.12	89,490
5	13,234	76,087	6.86	0.13	89,321
6	13,226	76,798	6.62	0.11	90,024
7	14,844	74,732	6.53	0.12	89,576
8	5,201	74,518	6.41	0.12	79,719
9	3,441	73,661	5.35	0.18	77,102
10	10,034	74,599	4.93	0.23	84,633
11	2,773	74,579	5.86	0.25	77,352
12	2,591	30,064	5.99	0.23	32,655

Total	**107,970**	**855,541**	**6.31**	**0.15**	**963,511**

Para explicar este comportamiento se realizó una gráfica (Figura 57) mostrando el comportamiento de los valores de plata y cobre a lo largo de los doce meses. Como se puede ver, estos dos elementos metálicos se comportan inversamente proporcional, es decir, cuando el valor de uno de ellos se incrementa, el otro disminuye.

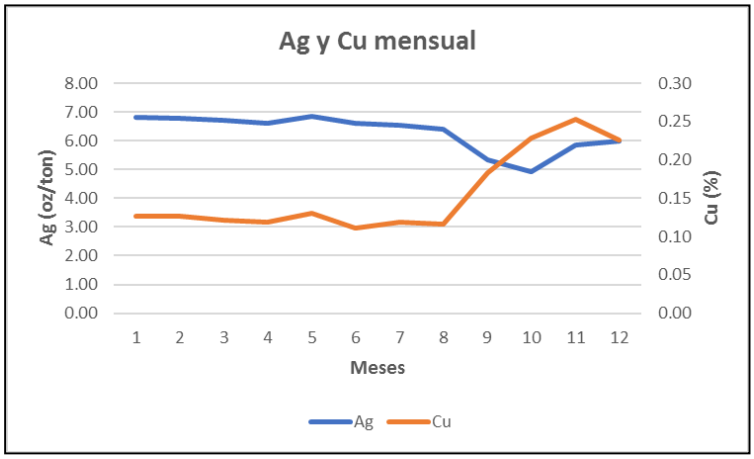


Figura 57. Comportamiento de ley de plata y cobre del plan de extracción

Por lo tanto, podemos decir que los últimos cuatro meses planificados presentan una disminución en la ley de plata y un incremento en el contenido de cobre.

No obstante, queda abierta la posibilidad de ampliar la presente investigación analizando desde el punto inicial que es el modelo de bloques para clasificar el material en un mayor número de rangos, y no solamente en mineral y desmonte. Podemos clasificar el mineral en cobre alto y cobre bajo por dar un ejemplo.

Otro factor que se analizó fue el origen de los materiales en cada mes del plan. Es decir, de qué stock se está extrayendo el material por cada mes. Para ello mostramos la Tabla 11 que muestra el tonelaje a extraer de cada stock en cada mes.

Tabla 10
Variables del ejemplo de la sección 4.4

Bloque	Periodo	Variable	Tonelaje	Valor
A1	1	X1	50	2
A2	1	X2	50	-2
A3	1	X3	50	4
A4	1	X4	50	3
B1	1	X5	50	-1
B2	1	X6	50	5
B3	1	X7	50	2
B4	1	X8	50	4
C1	1	X9	50	2
C2	1	X10	50	-1
C3	1	X11	50	3
C4	1	X12	50	3
A1	2	X13	50	2
A2	2	X14	50	-2
A3	2	X15	50	4
A4	2	X16	50	3
B1	2	X17	50	-1
B2	2	X18	50	5
B3	2	X19	50	2
B4	2	X20	50	4
C1	2	X21	50	2
C2	2	X22	50	-1
C3	2	X23	50	3
C4	2	X24	50	3
A1	3	X25	50	2
A2	3	X26	50	-2
A3	3	X27	50	4
A4	3	X28	50	3
B1	3	X29	50	-1
B2	3	X30	50	5
B3	3	X31	50	2
B4	3	X32	50	4
C1	3	X33	50	2
C2	3	X34	50	-1
C3	3	X35	50	3
C4	3	X36	50	3

Tabla 11
Producción por stock generada por el programa

Mes	Stocks (t)							Total Material (t)
	1	2	4	5	6	7	8	
1	13,672	7,543	21,510	23,650	10,537	9,669	1,189	87,770
2	31,094	40	14,901	14,647	5,245	9,702	6,256	81,885
3	42,510	-	11,249	-	11,295	2,475	16,457	83,986
4	54,072	-	7,924	1,579	-	13,568	12,347	89,490
5	23,097	21,036	13,358	-	194	7,490	24,145	89,320
6	15,673	14,012	12,647	-	18,720	25,627	3,345	90,024
7	21,832	27,991	1,690	-	-	-	38,063	89,576
8	26,723	22,872	-	-	-	-	30,124	79,719
9	48,195	26,966	-	-	-	-	1,942	77,103
10	84,630	-	-	-	-	-	-	84,630
11	77,352	-	-	-	-	-	-	77,352
12	32,655	-	-	-	-	-	-	32,655
Total (t)	**471,505**	**120,460**	**83,279**	**39,876**	**45,991**	**68,531**	**133,868**	**963,510**

Otro factor que podemos inferir analizando la Tabla 11 que otra de las causas por las que el secuenciamiento no logra cumplir las restricciones impuestas en los últimos periodos es debido a que se termina el material de varios stocks, quedando únicamente dos o un stock del cual seleccionar el material.

Esto se debe a que los stocks 1 y 7 tienen mayor tonelaje que los demás, por lo que terminan abasteciendo el plan de extracción los últimos meses solos.

Finalmente, valorizamos el plan con los parámetros de la sección 1 del presente capítulo.

La evaluación económica se realizará para cada periodo (mes) obtenido en el plan generado con el secuenciamiento.

Para realizar la evaluación económica, calcularemos el ingreso y el costo de cada periodo. El ingreso se calcula multiplicando el tonelaje de mineral del periodo, la ley de plata del mineral a extraer en el periodo, la recuperación metalúrgica de la plata y el precio de la plata. El costo se calcula sumando el producto del tonelaje total del periodo por el costo de minado, y el producto del tonelaje de mineral por el costo de procesamiento. El beneficio obtenido de cada periodo será la sustracción del ingreso y el costo.

Para el primer mes, según el plan que se puede apreciar en la Tabla 9, el cálculo seria el siguiente:

Datos del periodo 1:

Tonelaje de mineral: 77,377 t

Tonelaje de desmonte: 10,392 t

Tonelaje total: 87,769 t

Ley de plata: 6.82 oz/t

Recuperación metalúrgica: 0.25

Precio: 9.425 $/oz

Costo de minado: 2.0 $/t

Costo de procesamiento: 5.5 $/t

Ingreso: 77,377 t x 6.82 oz/t x 0.25 x 9.425 $/oz = 1,243,706 $

Costo: 87,769 t x 2.0 $/t + 77,377 t x 5.5 $/t = 599,115 $

Beneficio: 1,243,706 – 599,115 = 644,591 $

El mismo procedimiento realizaremos para cada periodo. En la Tabla 12 presentamos los resultados de todos los periodos secuenciados.

El ingreso total obtenido en el secuenciamiento es de 12.72 millones de dólares, el costo total sería de 6.63 millones de dólares y el beneficio sería de 6.08 millones de dólares. Sin embargo, si no se cumplieran los últimos cuatro meses del plan, debido a los niveles elevados de contenido de cobre, el beneficio total a obtenerse en los primeros 8 meses sería de 4.77 millones de dólares.

También mostramos una gráfica en la Figura 58 para ver el comportamiento de la ley de cobre y el beneficio. Se puede notar que el beneficio o utilidad es inversamente proporcional al contenido de cobre.

Tabla 12
Valorización económica del plan

Mes	Desmonte (t)	Mineral (t)	Leyes		Total Material (t)	Ingreso ($)	Costos ($)	Beneficio ($)
			OZAG (oz/t)	CU (%				
1	10,392	77,377	6.82	0.13	86,770	1,243,706	599,115	644,591
2	8,857	73,028	6.78	0.13	82,885	1,166,955	567,425	599,530
3	9,327	74,658	6.71	0.12	84,085	1,180,810	578,789	602,021
4	14,050	75,440	6.62	0.12	89,790	1,176,134	594,498	581,636
5	13,234	76,087	6.86	0.13	90,321	1,229,026	599,122	629,905
6	13,226	76,798	6.62	0.11	89,023	1,197,301	600,433	596,869
7	14,844	74,732	6.53	0.12	89,076	1,149,541	589,178	560,364
8	5,201	74,518	6.41	0.12	80,019	1,124,722	569,889	554,833
9	3,441	73,661	5.35	0.18	77,412	928,923	559,958	368,965
10	10,034	74,599	4.93	0.23	85,033	866,482	580,359	286,123
11	2,773	74,579	5.86	0.25	77,052	1,029,838	564,288	465,550
12	2,591	30,064	5.99	0.23	33,055	424,096	231,462	192,634
Total	**107,969**	**855,541**	**6.31**	**0.15**	**964,520**	**12,717,536**	**6,634,516**	**6,083,020**

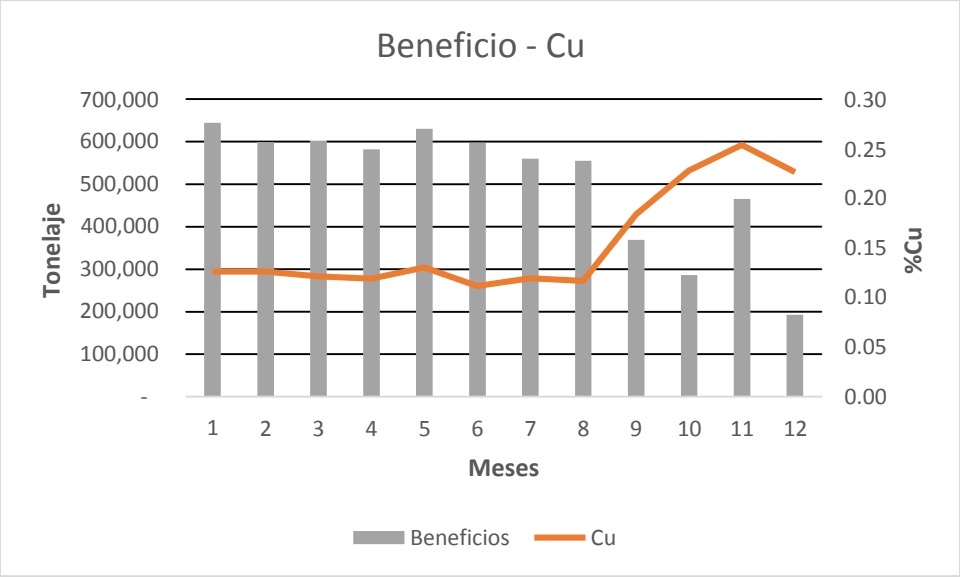


Figura 58. Gráfica mostrando el comportamiento del cobre y el beneficio

Conclusiones

1) Es factible realizar secuenciamiento de extracción de ocho stockpiles aplicando programación lineal entera mixta y Visual Basic.
2) Los parámetros de entrada subidos al programa de diseño de mina significaron el punto de partida del cálculo de la presente investigación.
3) Uno de los procedimientos más importantes para el presente estudio fue clasificar el material considerando los parámetros económicos.
4) Secuenciar a partir de polígonos en lugar de secuenciar a partir del mismo modelo de bloques ahorra el número de iteraciones en 93%.
5) El dimensionamiento del tamaño de la unidad de planificación es muy importante para generar un secuenciamiento eficaz.
6) La implementación del sub inventario donde sólo cargamos los 10 primeros polígonos por cada stock nos permitió reducir el número de variables del sistema de 594 variables a 80 variables.
7) El plan generado con la metodología presentada en esta investigación nos da un beneficio de USD 6,083,020, lo cual obtenemos al extraer 107,969 toneladas de desmonte y 855,541 toneladas de mineral.
8) Al visualizar los resultados espacialmente en el programa MineroSuite podemos hacer más operativos los polígonos y el mismo plan de extracción.

Recomendaciones

1) El éxito en la obtención de un plan de extracción de los stocks dependerá de la compatibilidad de los stocks con las restricciones impuestas.
2) Se debe ser muy riguroso al evaluar la data de entrada que se proveerá al sistema. Si no identificamos datos incorrectos en esta etapa, los cálculos posteriores resultarán errados.
3) El ancho mínimo de minado es un parámetro muy importante para obtener los polígonos de minado a partir del modelo de bloques.
4) A mayor dimensión de polígonos será menor el tiempo de ejecución del programa. Sin embargo, polígonos más grandes pueden significar una mayor dilución o una menor recuperación del mineral.
5) Se debe ser muy preciso al calcular el tamaño óptimo para la unidad de planificación que se utilizará en el algoritmo de secuenciamiento.
6) Se debe generar una copia de seguridad del inventario antes de empezar a realizar el secuenciamiento para no distorsionar la data inicial.
7) La valuación económica presentada se realizó con las condiciones de mercado, especialmente referente al precio, que se tenían al momento de realizar la presente tesis.
8) El plan se puede visualizar en cualquier software de diseño que soporte archivos DXF.

Bibliografía

Aires, D. d. (2011). Investigación de Operaciones.

Amaya, J., & Nancel-Penard, P. (Junio de 2010). *Secuencias optimas de extracción usando modelos matemáticos y HPC.* Obtenido de https://www.codelco.com/flipbook/codelcodigital5/pdfs/10_NT_3_UCHILE_CMM_AMTC_DELPHOS.pdf

Blom, M., & R. Pearce, A. (2018). *Short-Term Planning for Open Pit Mines: A Review.*

Guzman, R., & Ortiz, J. (2 de Enero de 2014). *Minería chilena.* Obtenido de https://www.mch.cl/informes-tecnicos/software-para-mineria-herramientas-para-disminuir-la-incertidumbre/#

Herrada, A. (2007). *Planeamiento de Minado y Control de Mineral.*

hiru.eus. (s.f.). Obtenido de https://www.hiru.eus/es/matematicas/promacion-lineal

Mora, A. (Octubre de 2016). *prezi.com.* Obtenido de https://prezi.com/ql3aagwj9ve/funcion objetivo

phpsimplex. (s.f.). Obtenido de https://www.phpsimplex.com/teoria_metodo_simplex.html

Programacion lineal. (Septiembre de 2018). Obtenido de https://www.programacionlineal.net/resolucion_grafica.html

Rancel, M. R. (s.f.). *Procedimientos Sub, funciones y programas con módulos.*

Rockcontent. (Abril de 2019). Obtenido de https://rockcontent.com/es/blog/que-es-un-lenguaje-de-programacion/

Sabater, J. P. (2014). *Manual Básico para empezar a trabajar con macros de Visual Basic para Excel.*

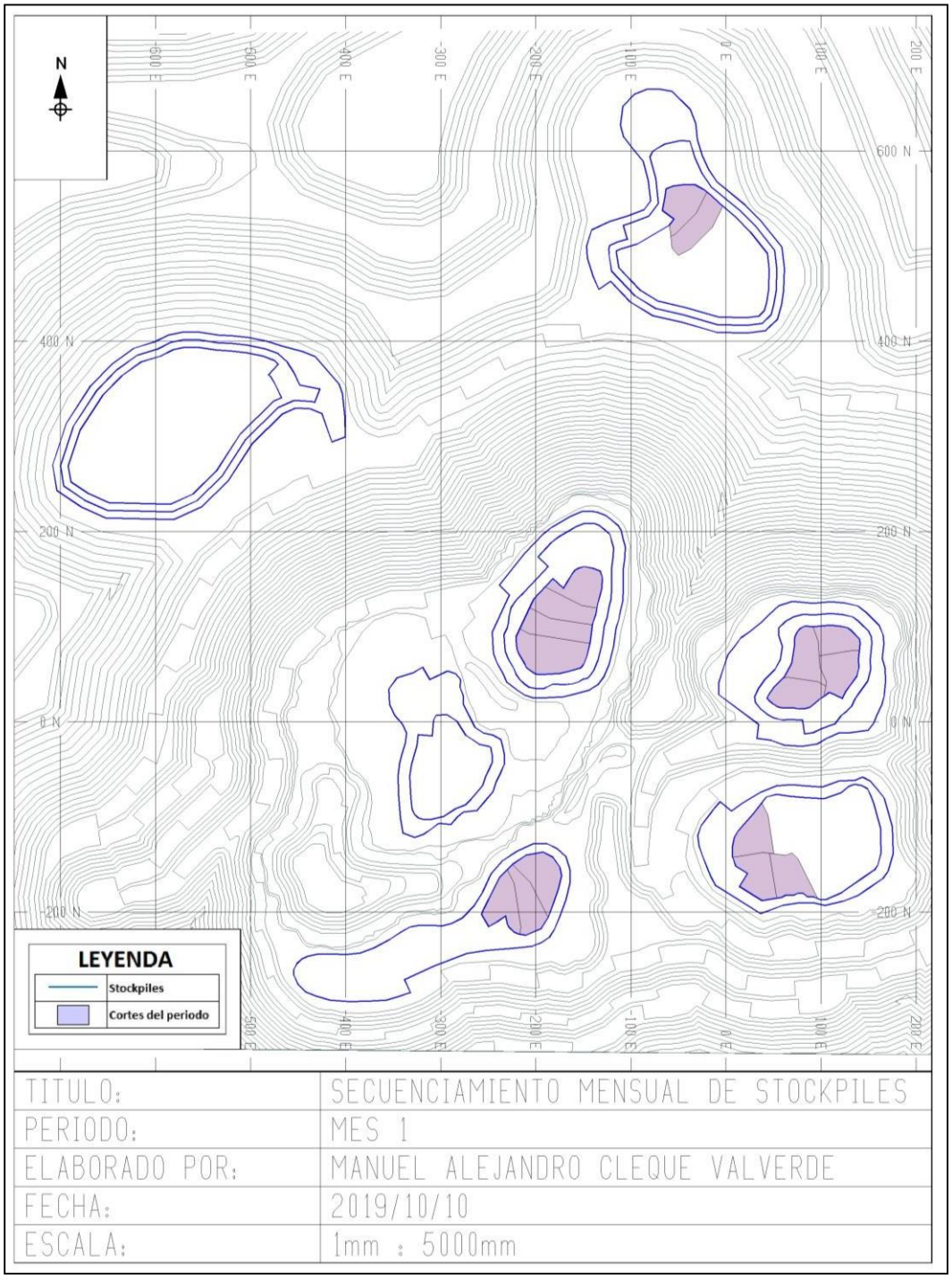
Salazar Lopez. (s.f.). *Ingenieria Industrial Online*. Obtenido de www.ingenieriaindustrialonline.com

Sinclair, A., & G. Blackwell. (2002). *Applied Mineral Inventory Estimation.*

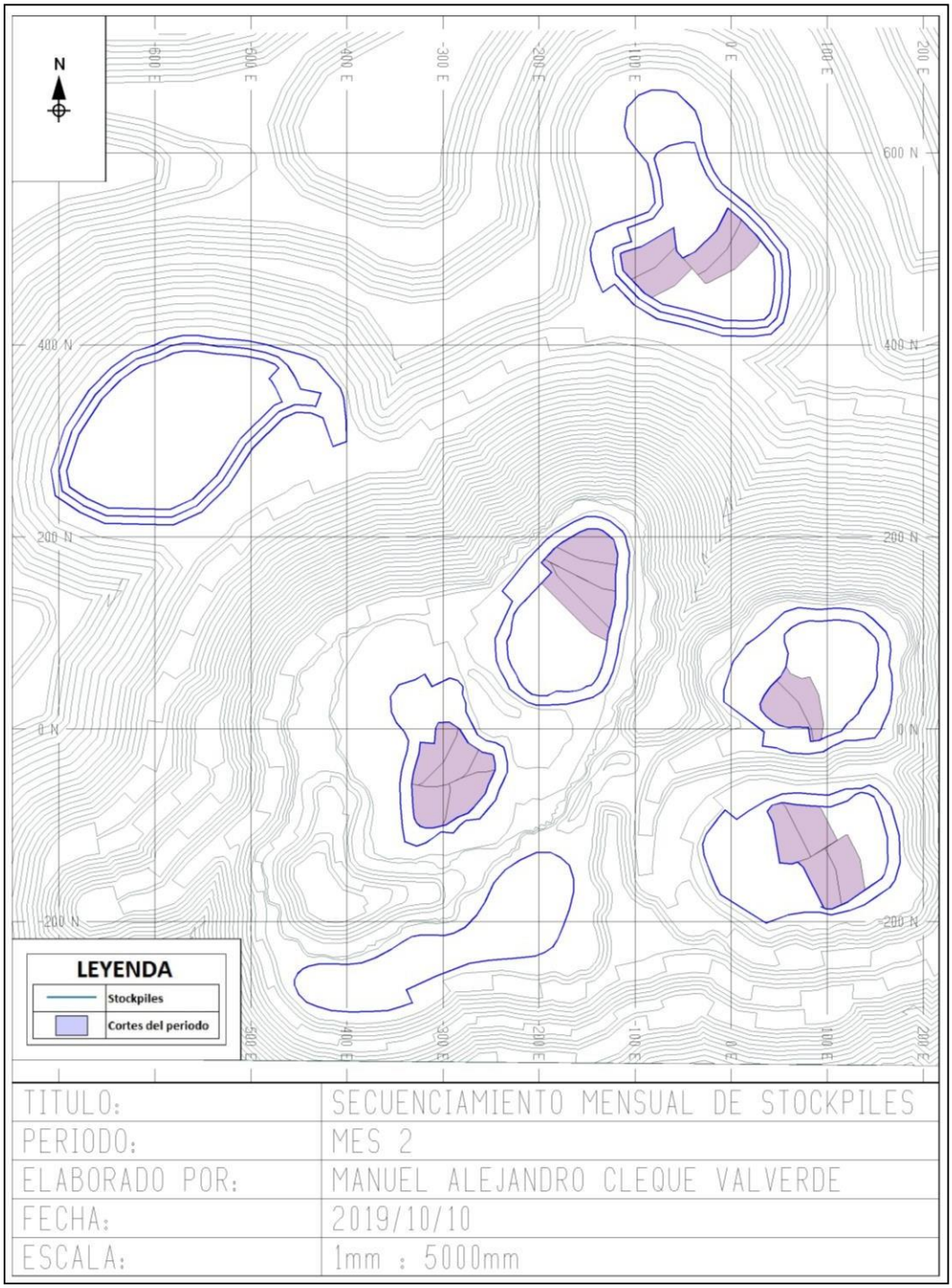
Taha, H. A. (2010). *Investigación de Operaciones.* Pearson.

Anexos

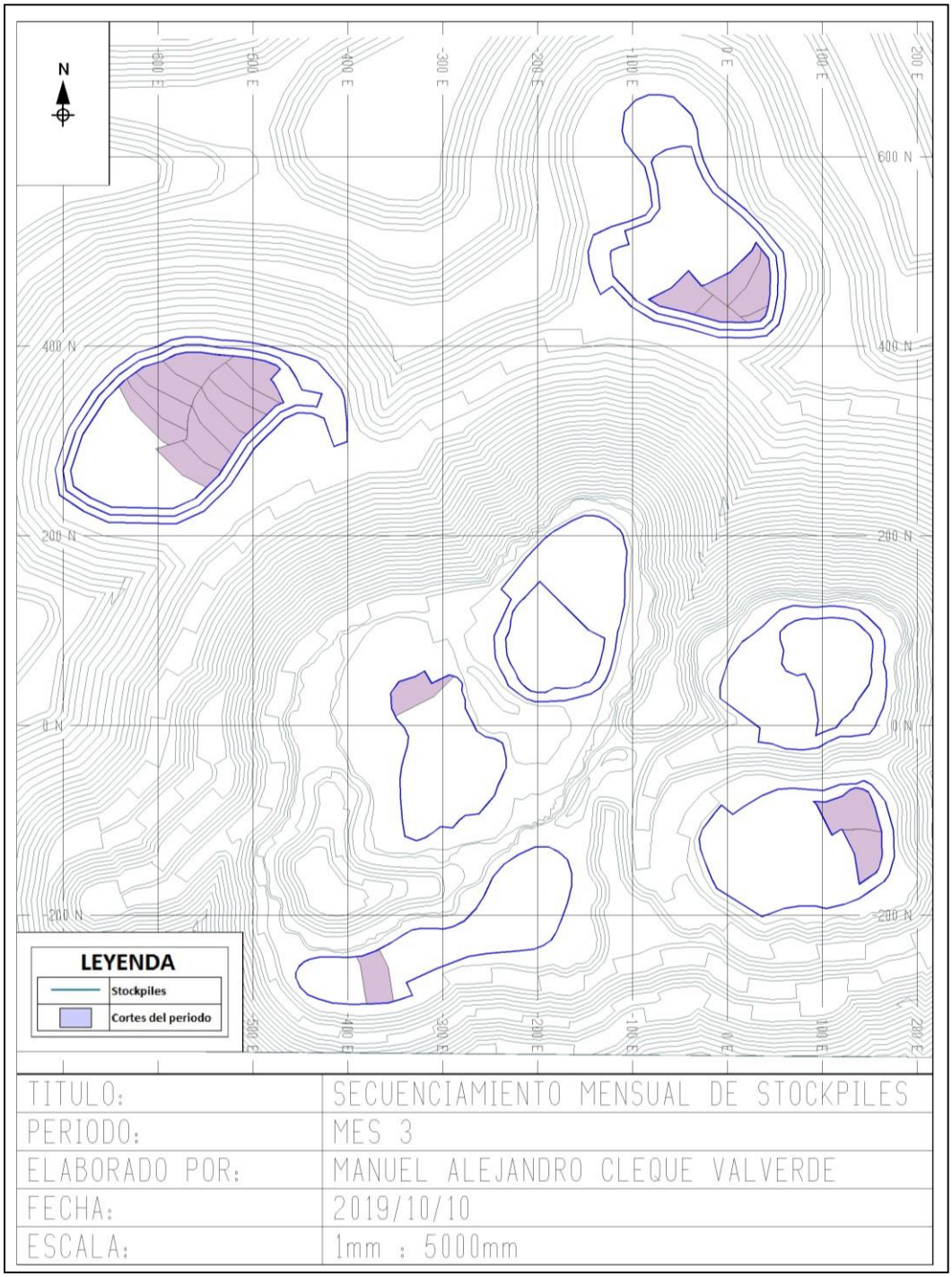
8.1. Anexo 1

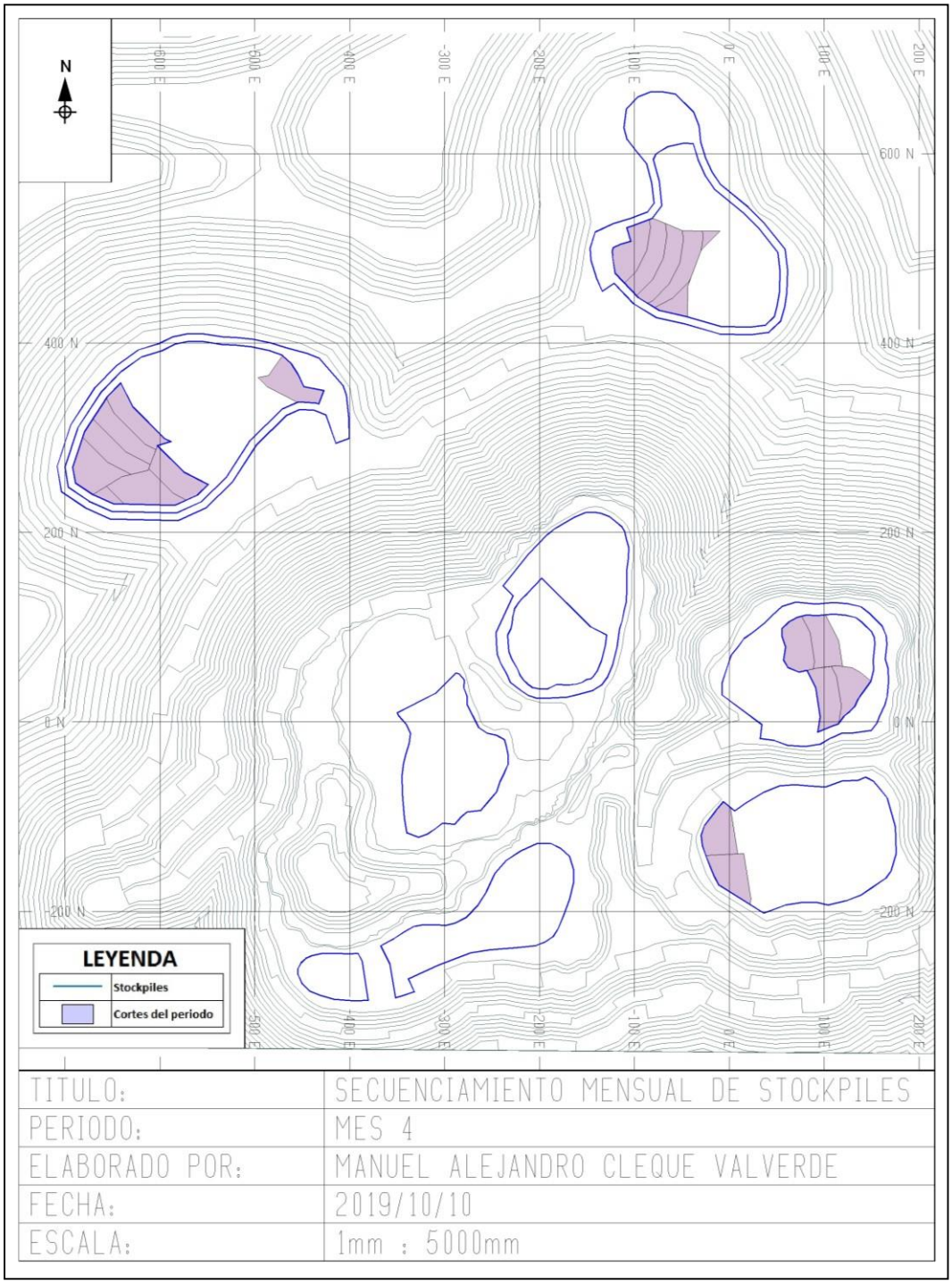


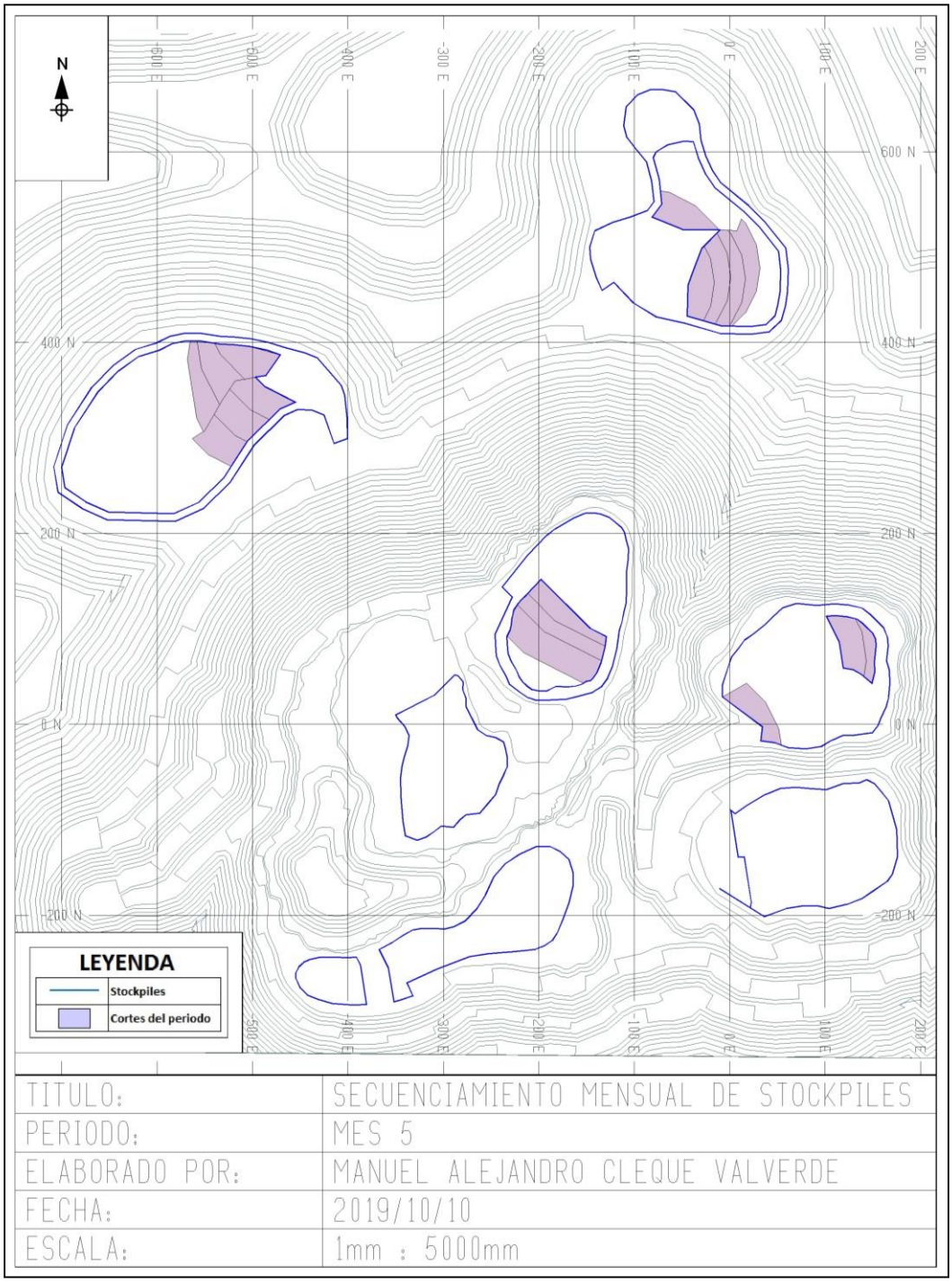
TITULO:	SECUENCIAMIENTO MENSUAL DE STOCKPILES
PERIODO:	MES 1
ELABORADO POR:	MANUEL ALEJANDRO CLEQUE VALVERDE
FECHA:	2019/10/10
ESCALA:	1mm : 5000mm



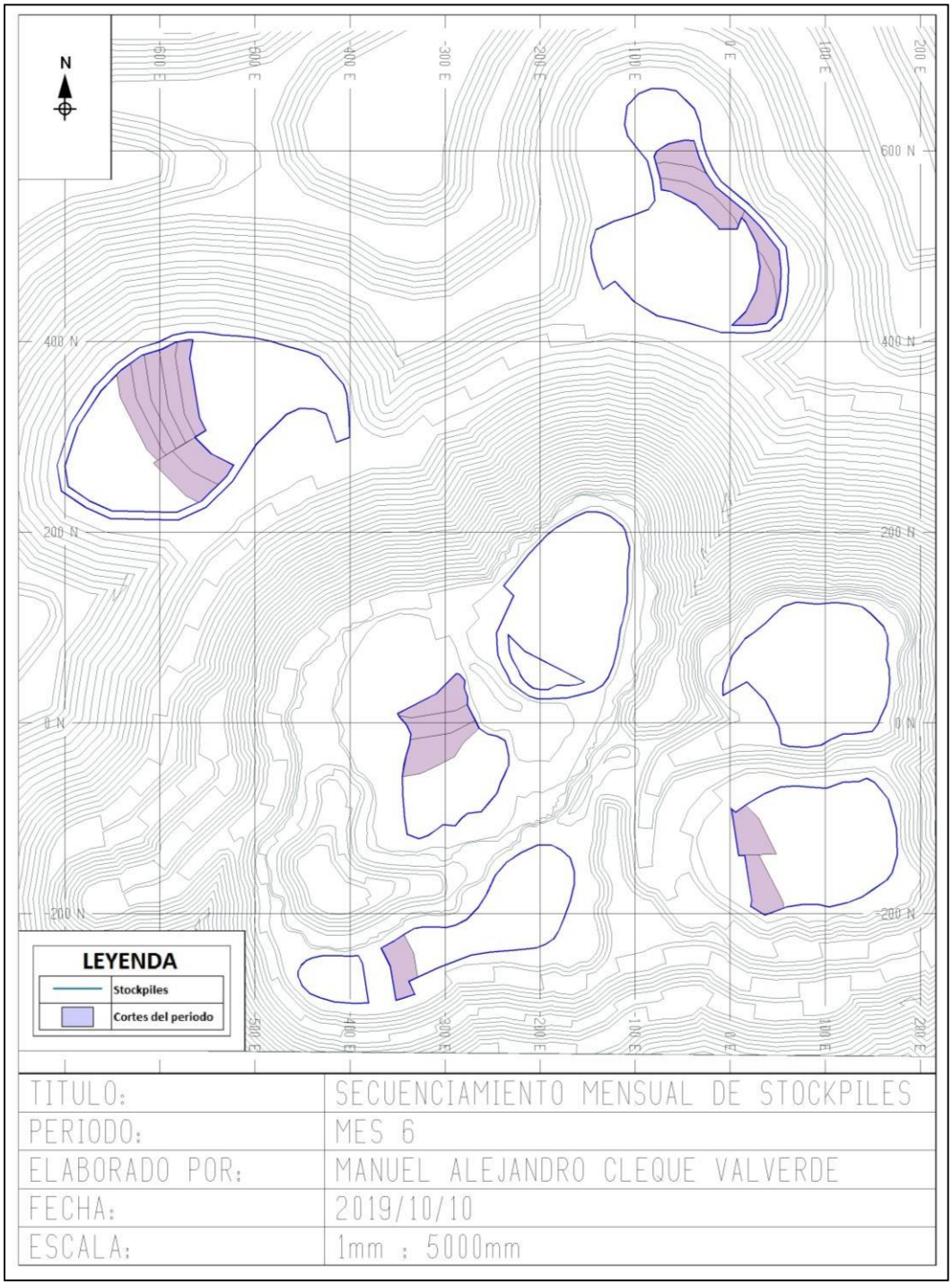
TITULO:	SECUENCIAMIENTO MENSUAL DE STOCKPILES
PERIODO:	MES 2
ELABORADO POR:	MANUEL ALEJANDRO CLEQUE VALVERDE
FECHA:	2019/10/10
ESCALA:	1mm : 5000mm


N
600 N
400 N
200 N
0 N
-200 N
LEYENDA
Stockpiles
Cortes del periodo
TITULO: SECUENCIAMIENTO MENSUAL DE STOCKPILES
PERIODO: MES 3
ELABORADO POR: MANUEL ALEJANDRO CLEQUE VALVERDE
FECHA: 2019/10/10
ESCALA: 1mm : 5000mm

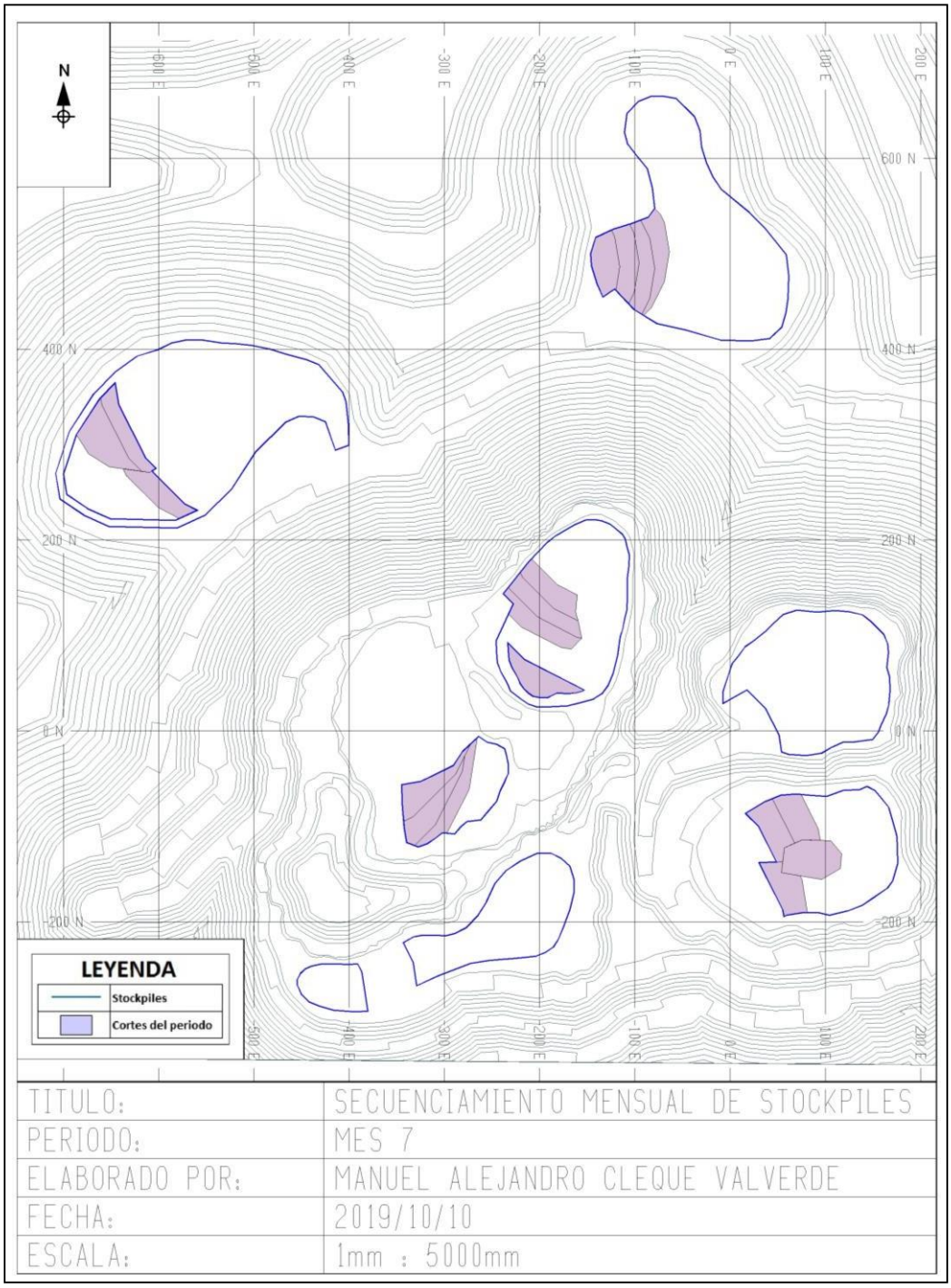

N
600 E
500 E
400 E
300 E
200 E
100 E
0 E
100 E
200 E
600 N
400 N
200 N
0 N
200 N
LEYENDA
Stockpiles
Cortes del periodo
TITULO: SECUENCIAMIENTO MENSUAL DE STOCKPILES
PERIODO: MES 4
ELABORADO POR: MANUEL ALEJANDRO CLEQUE VALVERDE
FECHA: 2019/10/10
ESCALA: 1mm : 5000mm

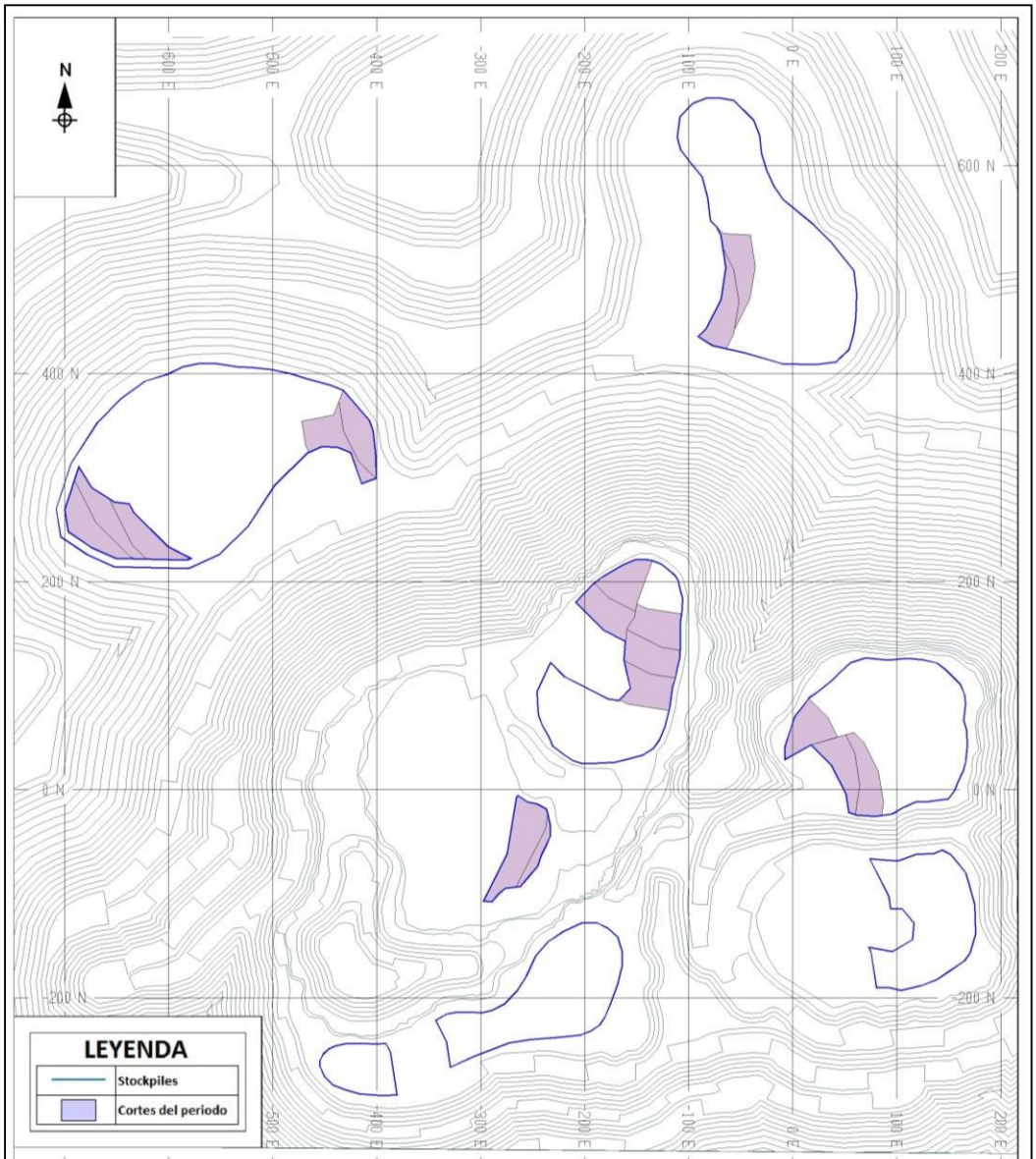


TITULO:	SECUENCIAMIENTO MENSUAL DE STOCKPILES
PERIODO:	MES 5
ELABORADO POR:	MANUEL ALEJANDRO CLEQUE VALVERDE
FECHA:	2019/10/10
ESCALA:	1mm : 5000mm

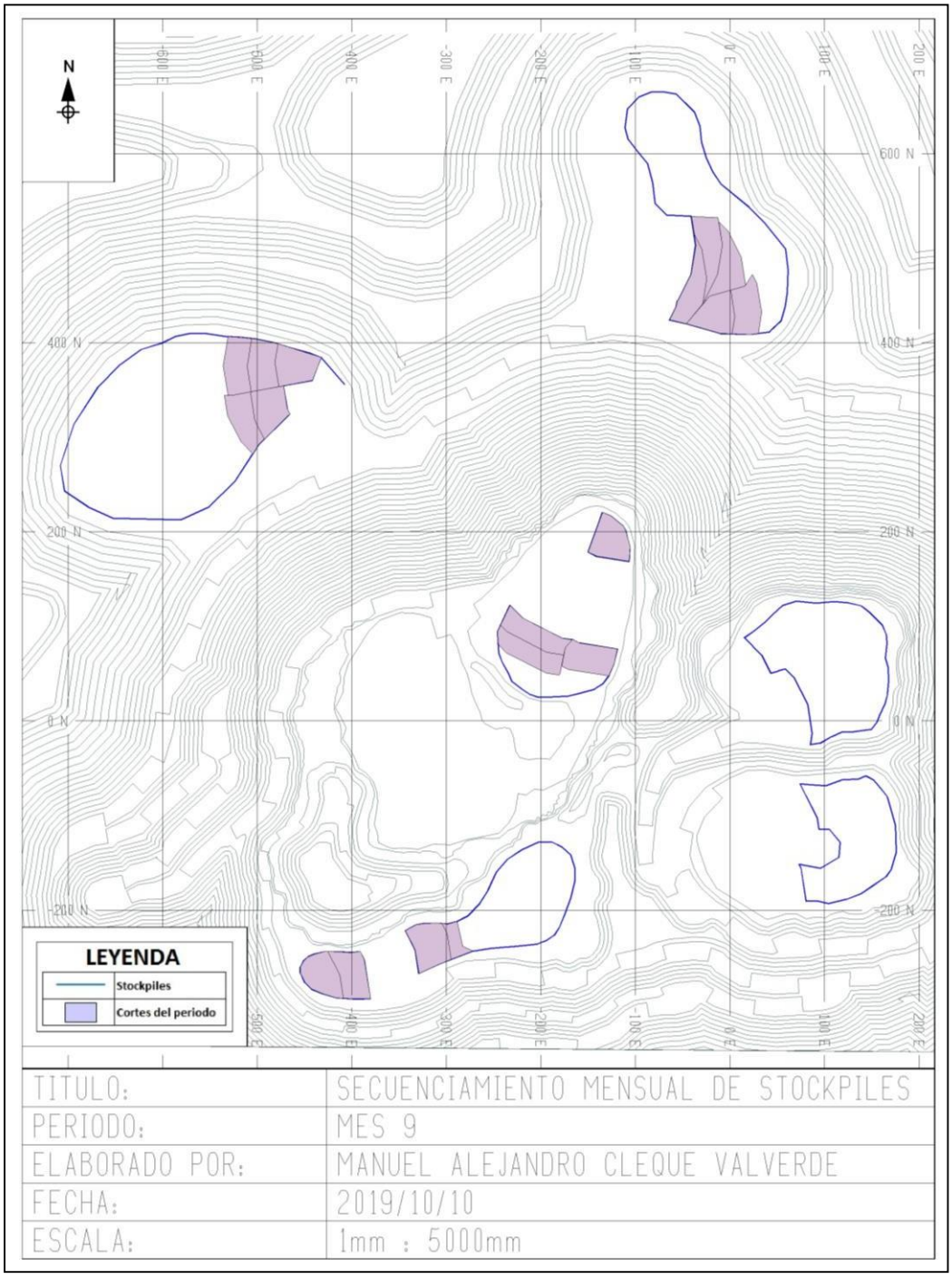


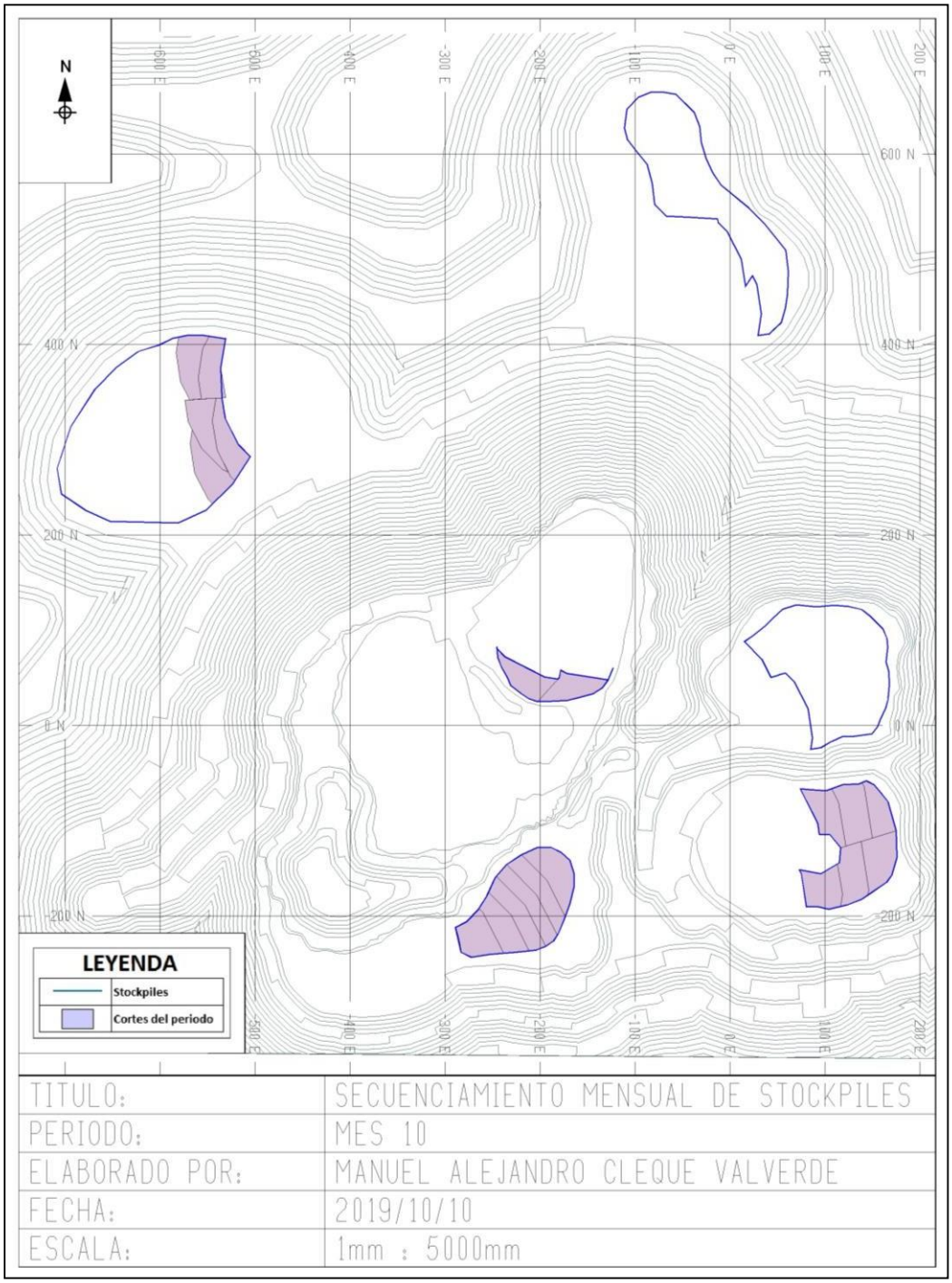
TITULO:	SECUENCIAMIENTO MENSUAL DE STOCKPILES
PERIODO:	MES 6
ELABORADO POR:	MANUEL ALEJANDRO CLEQUE VALVERDE
FECHA:	2019/10/10
ESCALA:	1mm : 5000mm

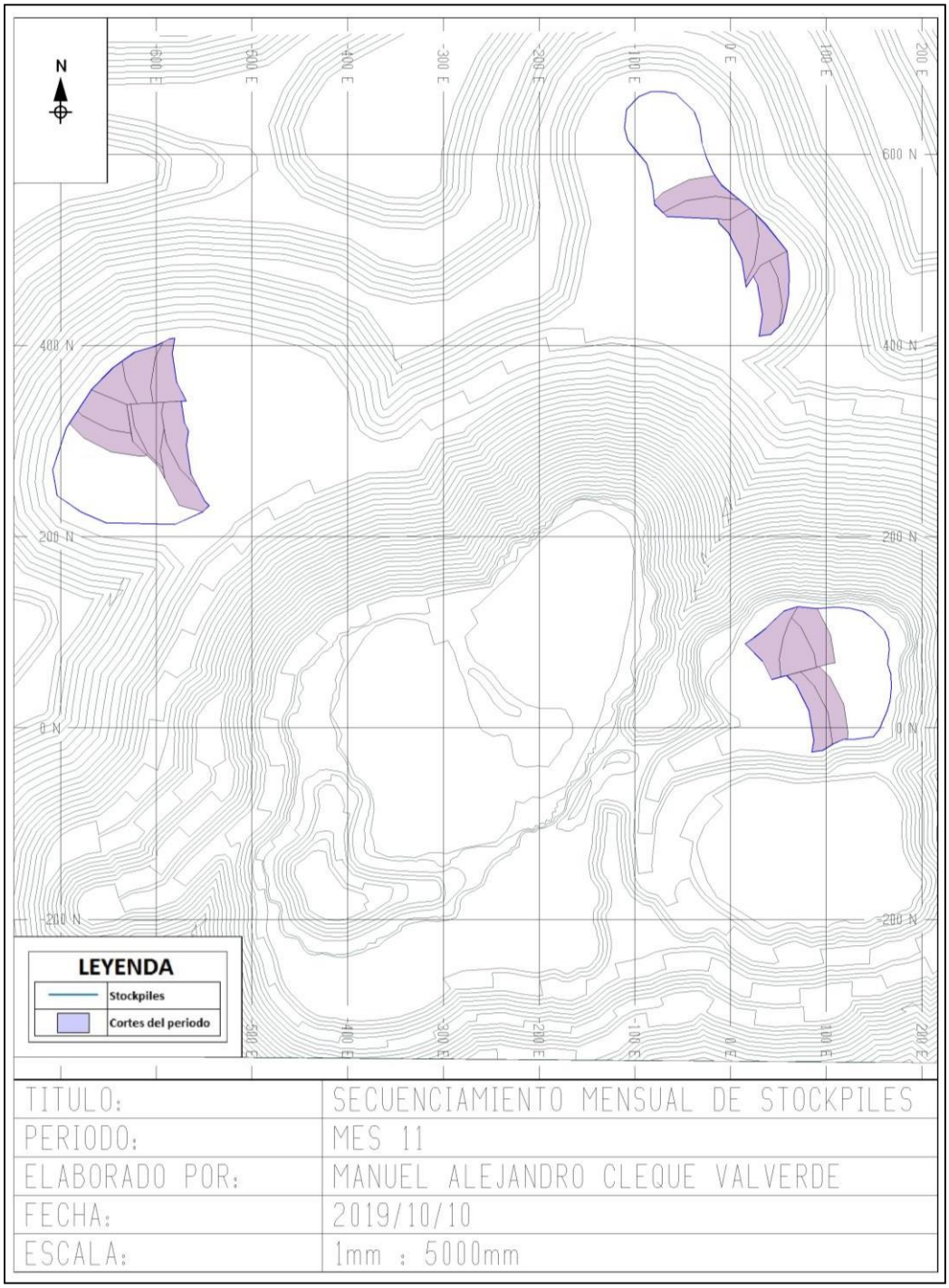

N
600 N
400 N
200 N
0 N
-200 N
LEYENDA
Stockpiles
Cortes del periodo
TITULO: SECUENCIAMIENTO MENSUAL DE STOCKPILES
PERIODO: MES 7
ELABORADO POR: MANUEL ALEJANDRO CLEQUE VALVERDE
FECHA: 2019/10/10
ESCALA: 1mm : 5000mm



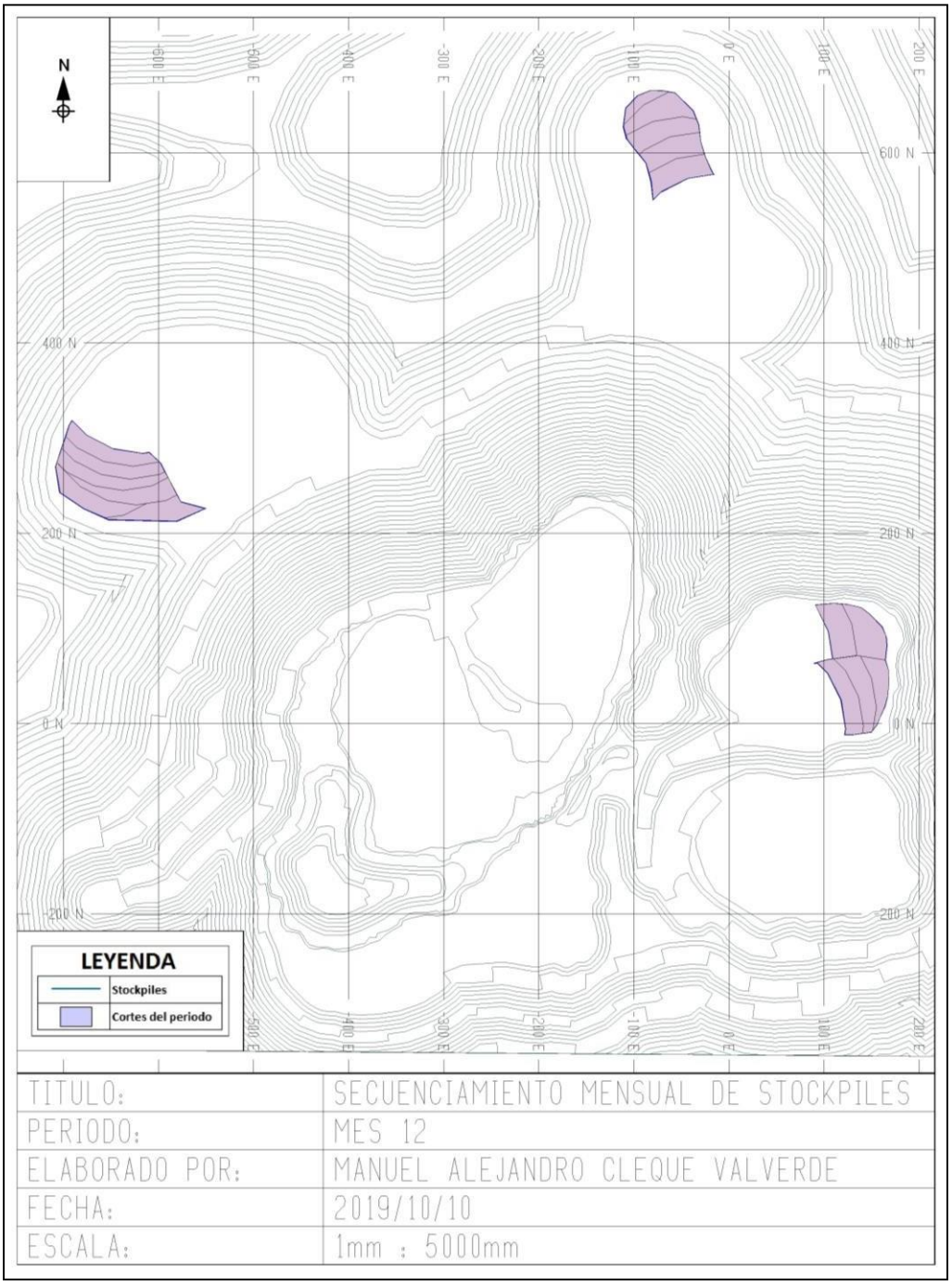
TITULO:	SECUENCIAMIENTO MENSUAL DE STOCKPILES
PERIODO:	MES 8
ELABORADO POR:	MANUEL ALEJANDRO CLEQUE VALVERDE
FECHA:	2019/10/10
ESCALA:	1mm : 5000mm


N
500 E
400 E
300 E
200 E
100 E
0 E
100 E
200 E
600 N
400 N
200 N
0 N
200 N
LEYENDA
Stockpiles
Cortes del periodo
TITULO: SECUENCIAMIENTO MENSUAL DE STOCKPILES
PERIODO: MES 9
ELABORADO POR: MANUEL ALEJANDRO CLEQUE VALVERDE
FECHA: 2019/10/10
ESCALA: 1mm : 5000mm


N
600 N
400 N
200 N
0 N
-200 N
LEYENDA
Stockpiles
Cortes del periodo
TITULO: SECUENCIAMIENTO MENSUAL DE STOCKPILES
PERIODO: MES 10
ELABORADO POR: MANUEL ALEJANDRO CLEQUE VALVERDE
FECHA: 2019/10/10
ESCALA: 1mm : 5000mm



TITULO:	SECUENCIAMIENTO MENSUAL DE STOCKPILES
PERIODO:	MES 11
ELABORADO POR:	MANUEL ALEJANDRO CLEQUE VALVERDE
FECHA:	2019/10/10
ESCALA:	1mm : 5000mm



TITULO:	SECUENCIAMIENTO MENSUAL DE STOCKPILES
PERIODO:	MES 12
ELABORADO POR:	MANUEL ALEJANDRO CLEQUE VALVERDE
FECHA:	2019/10/10
ESCALA:	1mm : 5000mm